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

Université Mouloud Mammeri de Tizi-Ouzou

Faculté de Génie Electronique et d'Informatique

Département Informatique

Mémoire de fin d'études

Pour l'obstination du diplôme Master en Informatique

Option : Conduite de Projet Informatique

Réalisé par : M^elle^ Idres Louiza

Thème

Système de détection d'intrusion hybride et hiérarchique pour les réseaux de capteur sans fil

Devant le Jury composé de :

M^r^ M. Daoui	Président
M^me^ R. Aoudjit	Promotrice
M^me^ Ben Hocini	Examinatrice
M^me^ M.Belkadi	Examinatrice

Promotion : 2011-2012.

Remerciements

Je tiens à remercier mon dieu, le tout puissant, de m'avoir donné le courage et la patience jusqu'à l'achèvement de ce travail.

J'exprime ma profonde reconnaissance et mes sincères remerciements à ma promotrice Mme AOUIDJIT Rachida pour sa disponibilité et son encadrement, ainsi que pour son soutien tout au long de l'année.

Egalement, je remercie les membres de jury, qui ont accepté d'évaluer ce modeste travail.

J'aimerais également remercier tous mes amis(e) de leurs soutient et aide et qui m'ont donnés la force pour résisté.

LOUIZA

Dédicaces

A mes chers parents,

A mes chers frères et ma sœur,

A toute la famille IDRES,

A tous mes chers(e) amis(e).

A tous ceux qui m'ont aidé de près ou de loin.

Je dédie ce travail.

Louiza

Liste des acronymes

ACA	Autonomous Component Architecture.
CH	Cluster Heads.
CPU	Central Processing Unit.
CSMA	Carrier Sense Multiple Access.
DRCL	Distributed Realtime Computing Laboratoire .
FN	False Negative.
FP	False Positif.
FPR	False Positive Rate.
HHIDS	Hybrid and Hierarchical Intrusion Detection System.
IDS	Intrusion Detection System.
IP	Inetrnet Protocol.
J-Sim	Java Simulator.
LEACH	Low-Energy Adaptive Clustering Hierarchy.
MAC	Message Authentification Code.
OSI	Open Systems Interconnexion.
RCSF	Réseau De Capteurs Sans Fil .
RUV	RUntime Virtual.
SOCSL	Self-Organized Criticality & Stochastic Learning Based IDS
TCL	Tool Command Line.
TDMA	Time Division Multiple Access.
TN	True Negative.
TP	True Positif .
TPR	True Positive Rate.
WSN	Wireless Sensor Network.

Table des matières

Table des acronymes 3

Table des matières 4

Introduction générale 11

Chapitre I :Généralité sur les RCSF

1. Introduction: 13
2. Les capteurs sans fil: 13
3. Définition d'un réseau de capteur sans fil : 14
4. Les caractéristiques des RCSF: 15
5. Les domaines d'application des RCSF : 17

5.1. Domaine militaire : 17

5.2. Domaine environnemental : 17

5.3. Domaine médical : 17

5.4. Domaine commercial: 17

6. La communication dans les RCSF : 17

6.1. les couches protocolaires 18

6.2. Les niveaux de gestion dans RCSF : 19

7. Conclusion: 20

Chapitre II : La sécurité dans les RCSF

1.Introduction: 21

2. Les vulnérabilités de sécurité dans les RCSF : 21

2.1.Limitation de ressources: 21

2.2.La communication non fiable: 22

2.3.Fonctionnement sans surviellance: 22

3.Les objectifsde la sécurité : 23

3.1.L'authentification: 23

3.2.L'intégrité : 23

Table des matières

3.3.La confidentialité : 23
3.4La disponibilité : 23
3.5.Le contrôle d'accès : 24
3.6.Non-répudiation : 24
4.Les mécanismes de sécurité dans les RCSF : 24
4.1.La cryptographie : 24
4.1.1.Le chiffrement : 24
4.1.2.La signature digitale : 25
4.1.3.La fonction de hachage : 25
4.1.4.Le code d'authentification des messages : 25
4.2.La gestion de clés : 25
5.Les attaques contre les RCSF: 26
5.1.La classification selon l'origine : 26
5.2.La classification selon la nature : 27
6.L'affectation des attaques ou différentes couches dans les RCSF : 27
6.1.Les attaques de la couche physique : 27
6.2.Les attaques de la couche liaison : 28
6.3.Les attaques de la couche routage : 28
6.4.Les attaques de la couche application : 31
7.Conclusion: 32

Chapitre III :Les systémes de détection d'intusion dans les RCSF

1. Introduction : 33
2. Système de détection d'intrusion : 33
3. Les composants d'un IDS : 33
4. Les propriétés des IDSs dans les RCSF : 35
5. L'architecture des systèmes de détection d'intrusion dans RCSF : 36
6. Les techniques de détection d'intrusion : 37
6.1. La détection par scénarios : 37

6.2. La détection par comportement : 38
6.3. La détection par spécifications : 38
7. Les modèles d'IDS existants pour RCSF : 38
7.1. Auto organisée et criticité stochastique d'apprentissage : 38
7.2. Les IDS décentralisés : 39
7.3. L'algorithme de détection des attaques de routage: 39
8. Conclusion : 40

Chapitre IV : le système de détection d'intrusion proposé

1.Introduction : 41
2.L'idee de base : 41
3.L'architectur earborescente de HHIDS: 42
3.1.Au niveau des clusters head : 42
3.2.Au niveau de la station de base : 42
4.L'architecture interne de HHIDS : 42
4.1.Module de détection d'intrusion : 43
4.2.Module de prise de décision : 43
5.Les techniques de detection utilisees : 43
5.1.Régles utilisées pour la détection : 44
6.Le schema de securisation de HHIDS : 45
6.1.Format des paquets à envoyer : 45
6.2.Les services de HHIDS : 45
6.2.1. Règle d'intervalle : 46
6.2.2. Règle d'intégrité : 46
6.2.3. Gamme de transmission par radio : 46
6.2.4. Règle de répétition : 47
6.2.5. Règle de blocage : 47
6.2.6. Règle de fraicheur : 47
7.Deroulementde HHIDS : 46

7.1.Au niveau du cluster head : 46
7.2.Au niveau de la station de base : 47
7.3.Détection de l'attaque de collision : 47
7.4.Détection de l'attaque de rejoue : 48
8.Conclusion : 48

Chapitre V : Simulation et étude des résultats

1. Introduction : 54
2. Présentation de J-Sim: 54
2.1. Types de nœuds : 55
2.1.1. Les nœuds Target : 56
2.1.2. Le nœud Sink : 56
2.1.3. Les nœuds Sensor : 57
3. Implémentation deHHIDS : 58
3.1. La structure des paquets échangés : 58
3.2. Quelques fonctions de HHIDS : 59
3.3. Implémentation des attaques : 61
4. Métriques d'évaluation de HHIDS : 62
4.1. Paramétrage de la simulation 63
4.2. Résultats et interprétation : 63
4.2.2. Consommation d'énergie : 66
5. Conclusion : 67
Conclusion générale : 68
Référence bibliographique : 70
Annexe A : Protocol LEACH : 74
Annexe B : l'installation de J_Sim : 77

Liste des figures

Figure I. 1 : Les composants d'un nœud capteur……....13
Figure I. 2 : Architecture de communication d'un RCSF……....14
Figure I. 3: Pile protocolaire des réseaux de capteurs sans fil..... …17

Figure II.1:Fonctions de la gestion de clés 25
Figure II. 2 :l'attaque d'inondation 28
Figure II. 3:l'attaque de trou ver.....29
Figure II. 4:l'attaque de trou noir.....29
Figure III. 2 : l'architecture hiérarchique. 36
Figure IV. 3 :l'architecture de HHIDS.....44
Figure IV. 4: Le fonctionnement interne du module de détection d'intrusion. 45
Figure V. 1: Connexions entre composants dans J-Sim.....55
Figure V. 2: la communication des nœuds dans J-Sim 56
Figure V. 3: Vue interne d'un nœud Target.....56
Figure V. 4: Vue interne d'un nœud Sink 57
Figure V. 5:Architecture interne d'un nœud Sensor dans J-Sim. 58
Figure V. 6:Histogramme représentant les paquets reçu par la station de base.....64
Figure V. 7:la durée de vie des nœuds sans HHIDS. 66
Figure V. 8:la durée de vie des nœuds avec HHIDS.....67

Liste des tableaux

Tableau. II.01 : Attaques contre les RCSF……32

Tableau. V.01 : Paramètres du la simulation……63

Liste des graphes

Gph.1 V-1: le taux de faux positif.....67

Gph.1 V-2 : le taux de faux négatif. 67

Introduction générale

Au cours de ces dernières années, le développement technologique des réseaux de communication sans fil, a connu un essor important grâce aux avancées technologiques dans divers domaines, telles que la micro-électronique et la miniaturisation. C'est ainsi que de nouvelles voies d'investigation ont été ouvertes avec l'émergence des réseaux de capteurs sans fil qui sont composés de petits dispositifs électroniques autonomes, équipés de capteurs et capables de communiquer entre eux via des liaisons sans fil.

En plus ils collaborent entre eux pour former un réseau de capteurs sans fil capable de surveiller une région ou un phénomène d'intérêt, de fournir des informations utiles par la combinaison des mesures prises par les différents capteurs et de les communiquer ensuite via le support sans fil à un centre de contrôle distant.

Les nœuds capteurs sont conçus pour être déployés d'une manière dense dans des endroits hostiles et difficiles d'accès, d'où la nécessité de limiter au maximum leurs dimensions physiques qui s'obtiennent impérativement au détriment des capacités de calcul, de traitement et de ressources énergétiques.

En raison de leur déploiement en environnements ouverts, de leurs ressources limitées et de la nature du medium de transmission, les réseaux de capteurs doivent faire face à de nombreuses attaques. Plusieurs mécanismes ont été proposés pour assurer la sécurité dans ces réseaux tels : la cryptographie symétrique, l'authentification et la sécurité des protocoles de routage. Mais ces mécanismes sont pour la plupart hautement spécialisés pour une attaque précise, d'où la nécessité d'utilise les systèmes de détection d'intrusion (IDS) [49].

Un système de détection d'intrusion est un processus de contrôle et d'analyse des évènements dans un réseau afin de reconnaitre des intrusions ou des tentatives d'intrusions par la reconnaissance de comportements anormaux ou d'attaques. Les techniques de détection se basent principalement soit sur la détection comportementale qui consiste à comparer le comportement observé du système cible aux comportements normaux espérés, ou sur la détection par scénarios qui consiste à faire une comparaison entre les comportements observés et les scénarios d'attaques prédéfinis et enfin la technique hybride qui rassembles ces deux dernières.

Dans ce travail nous avons développé un système de détection d'intrusions qui se base sur la technique hybride pour la détection d'intrusion, ainsi que sur l'architecture hiérarchique pour l'organisation des nœuds de réseau sous forme de clusters (Hybrid and Hierarchical Intrusion Detection System : HHIDS). Dont l'objectif de HHIDS est d'offrir un niveau de sécurité acceptable tout en minimisant le surcoût en termes de ressources.

Pour mieux cerner l'objectif de notre étude, nous introduirons dans un premier chapitre les RCSF, leurs caractéristiques, leurs domaines d'applications ainsi que leurs architectures. Dans le deuxième chapitre, nous présenterons la problématique de la sécurité dans les RCSF en exposant les différents types d'attaques qui peuvent viser le réseau. Nous définirons aussi les services de base de la sécurité et les mécanismes qui les vérifient. Dans le troisième chapitre, nous présenterons les systèmes de détection d'intrusion dans les réseaux de capteurs, leurs propriétés, les techniques de détection d'intrusions, ainsi que leurs architecture et composants et nous terminerons par quelques modèles de système de détection d'intrusion existants. Dans le quatrième chapitre, nous expliquerons notre démarche de conception d'un nouveau système de détection d'intrusion, nous commencerons par l'étude de l'architecture utilisés pour l'organisation des nœuds de réseau et le fonctionnement du système de détection d'intrusion proposé. Le cinquième chapitre, quant à lui, nous permettra d'exposer les résultats d'implémentation et de tests de simulation de notre solution, précédé par une présentation des outils nécessaires pour notre réalisation à savoir le simulateur j-Sim et le langage de script TCL. Enfin, nous clôturons par une conclusion et des perspectives.

Chapitre I : Généralités sur les RCSF

1. Introduction:

Depuis quelques décennies, le besoin d'observer et de contrôler des phénomènes physiques tels que la température, la pression ou encore la luminosité et le degré de pollution de l'air, est essentiel pour de nombreuses applications industrielles et scientifiques. Les nœuds utilisés doivent être autonomes, d'une taille miniature et peuvent être déployés d'une manière dense et aléatoire dans le champ surveillé.

Aujourd'hui , grâce aux développements technologiques tels que la miniaturisation des composants électroniques, la diminution des coûts de fabrication, l'augmentation des performances et des capacités de stockage d'énergie et de calcul, et grâce aux récents progrès des technologies sans fil, de nouveaux produits exploitant des réseaux de capteurs sans fil sont employés pour récupérer ces données environnementales.

Ce chapitre présente un état d'art sur les réseaux de capteurs, en commençant par la description d'un nœud capteur et ses composants, et terminant par définir les réseaux de capteur sans fil , leurs caractéristiques, leur architecture de communication, ainsi que leurs domaines d'application et leurs contraintes.

2. Les capteurs sans fil:

Un capteur sans fil est un petit dispositif électronique intelligent capable de mesurer une valeur physique environnementale (température, lumière, pression, etc.), et de la communiquer à un centre de contrôle via une station de base.

Chaque capteur assure trois fonctions principales [1] [2] [3] : la collecte, le traitement et la communication de l'information vers un ou plusieurs points de collecte. Il est constitué de quatre unités principales comme le montre la Figure I.1:

- **Unité de capture** :

  Elle est constituée de deux composants, un dispositif qui intercepte les données du monde physique et les transforme en signaux analogiques et un convertisseur analogiques/numériques, qui transforme ces signaux analogiques en signaux numériques compréhensible par l'unité de traitement.

- **Unité de traitement :**

Elle est composée d'un microprocesseur ou d'un microcontrôleur associé généralement à une unité de stockage. Elle est chargée d'exécuter les protocoles de communication, comme elle peut aussi effectuer des semi-traitements sur les données captées.

- **Unité de transmission :**

Elle est responsable des émissions et réceptions des données sur un médium sans fil. Elle se base sur les technologies sans fil à faible portée de communication tel que Bluetooth, WiFi et Zigbee.

- **Unité d'énergie :**

Elle est responsable de la gestion de l'énergie et de l'alimentation de tous les composants du capteur. Elle consiste, généralement, en une batterie qui est limitée et irremplaçable, ce qui a rendu l'énergie comme principale contrainte pour un capteur.

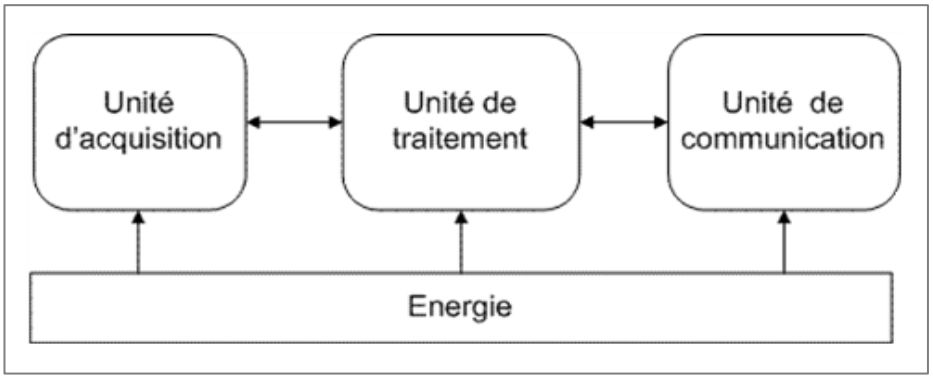


Figure I. 1 : Les composants d'un nœud capteur.

3. Définition d'un réseau de capteur sans fil :

Un réseau de capteurs sans fil (RCSF) ou Wireless Sensor Network(WSN) est constitué d'un ensemble de dispositifs très petits, nommés nœuds capteurs dispersés dans une zone géographique appelée champs de captage [1] [4]. Chacun de ces nœuds a la capacité de collecter les données et les router vers la station de base (dit "sink" en anglais ou puits). Ce dernier récupère les informations remontées par les différents capteurs et les transmet au centre de traitement via internet ou par satellite [5] .la figure I.2 présente un exemple de réseaux de capteurs sans fil.

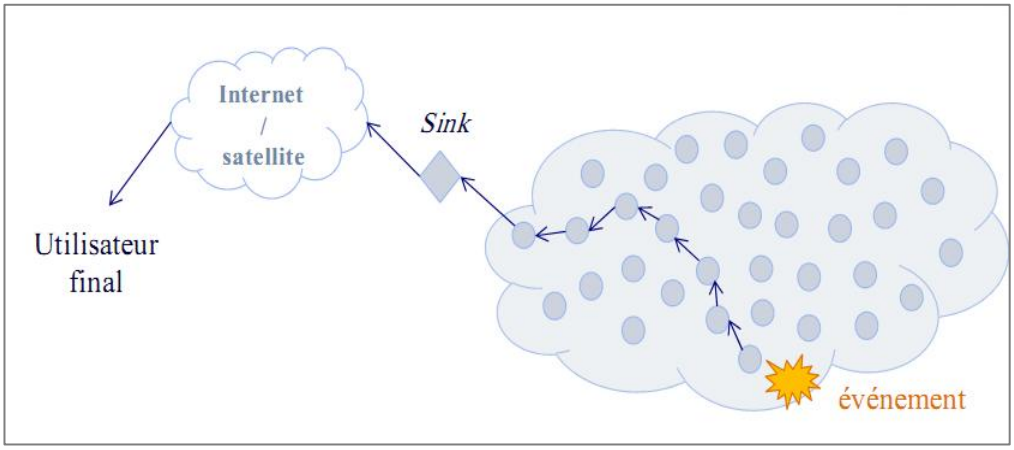


Figure I. 2 : Architecture de communication d'un RCSF

4. Les caractéristiques des RCSF:

Les principales caractéristiques qui différencient les RCSF par rapport aux autres réseaux sont illustrées ci-dessous [4] [3] [1]:

- **Les ressources limitées :**

  Les nœuds, qui composent le RCFS, sont de très petite taille jusqu'à l'ordre d'une poussière (smart dust), suivant leur utilité. Par conséquent, leurs capacités de traitement, de stockage, de communication et d'énergie sont très limitées.

- **La durée de vie limitée**:

  L'alimentation de chaque nœud est assurée par une source d'énergie limitée et généralement irremplaçable à cause de l'environnement hostile où il est déployé. Cette contrainte énergétique influe directement sur la durée de vie des capteurs, ainsi sur celle du réseau entier. De ce fait, la durée de vie d'un RCSF dépend fortement de la conservation d'énergie au niveau de chaque nœud.

- **La topologie dynamique :**

  La topologie d'un RCSF peut être dynamique, ceci est due à la mobilité des nœuds qui peuvent s'attacher à des objets mobiles, ou due à l'ajout ou à la suppression d'un nœud après le déploiement pour élargir le réseau ou pour remplacer les nœuds décédés ou défaillants**.**

- **La densité de déploiement**:

  Le nombre de nœuds capteurs peut atteindre des millions de nœuds pour permettre une meilleure granularité de surveillance. De plus, si plusieurs nœuds capteurs se retrouvent dans une région, un nœud défaillant pourra être remplacé par un autre. Cependant, la densité de déploiement donne naissance à des challenges pour la

communication entre les nœuds. En effet, elle provoque des collisions ou des endommagements des paquets transmis.

- **L'absence d'adressage fixe des nœuds**:

Les nœuds dans les réseaux sans fil classiques sont identifiés par des adresses IP. Cependant, cette notion n'existe pas dans les RCSF. Ces derniers utilisent un adressage basé sur l'attribut du phénomène capté, on parle donc de l'adressage basé attribut. En effet, les requêtes des utilisateurs ne sont pas généralement destinées à un seul nœud, mais plutôt, à un ensemble de nœuds identifiés par un attribut.

- **La limitation de ressources physiques** :

A cause de la miniaturisation des composants électroniques, les performances des nœuds capteurs sont limitées. Par conséquent, les nœuds capteurs collaborent en traitant partiellement les mesures captées et envoient seulement les résultats à l'utilisateur. Une autre conséquence, ces limitations imposent des portées de transmission réduites contraignant les informations à être relayées.

- **La sécurité physique limitée** :

Les réseaux de capteurs sans fil mobiles sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.

- **L'absence d'infrastructure** :

Les réseaux ad hoc, en général, se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.

- **La scalabilité** :

Les réseaux de capteurs engendrent un très grand nombre de capteurs, ils peuvent atteindre des milliers voir des millions de capteurs. Le défi à relever par les RCFS est d'être capable de maintenir leurs performances avec ce grand nombre de capteurs.

- **La bande passante limitée** :

En raison de la puissance limitée, les nœuds capteurs ne peuvent pas supporter des débits élevés. Ces différentes caractéristiques permettent aux réseaux de capteurs d'envahir rapidement plusieurs domaines d'applications présentés dans ce qui suit.

5. Les domaines d'application des RCSF :

Les réseaux de capteur sans fil pénètrent de plus en plus dans notre vie quotidienne. Ils sont employés dans plusieurs domaines d'applications ou ils peuvent offrir les meilleures contributions parmi lesquels nous citons :

5.1. Domaine militaire :

Un réseau de capteurs peut être déployé dans un endroit stratégique ou hostile, afin de surveiller les mouvements des forces ennemies, ou analyser le terrain avant d'y envoyer des troupes (détection des armes chimiques, biologiques ou radiations) [2] [6].

5.2. Domaine environnemental :

Dans ce domaine, les capteurs peuvent être exploités pour détecter les catastrophes naturelles (feux de forêts, tremblements de terre, …), traquer les mouvements des animaux et surveiller les conditions d'environnement qui affectent les récoltes, les stocks et tout autre système d'agriculture. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.)[3][2].

5.3. Domaine médical :

L'utilisation des réseaux de capteurs dans le domaine de la médecine peut apporter une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité, facilitant ainsi le diagnostic de maladies grâce à des micro-capteurs qui pourront être ingérés ou implantés sous la peau. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, ...) chez les personnes handicapées ou âgées [7] [2].

5.4. Domaine commercial:

Les capteurs peuvent être utilisés pour le contrôle environnemental des bâtiments, pour permettre une meilleure gestion des ressources à faibles coûts, la surveillance de l'état du matériel, le contrôle et l'automatisation des processus d'usinage [1] [3].

6. La communication dans les RCSF :

En prenant comme référence le modèle OSI (Open Systems Interconnexion : modèle de référence de base) pour la représentation en couche de la pile de protocoles de communication ; nous remarquons qu'il y a seulement cinq couches dans les réseaux de capteurs comme illustré dans la figure I.3.

Cette pile est également caractérisée par trois niveaux qui intègrent dans les protocoles des différentes couches la prise en compte de l'énergie consommée, de la mobilité des nœuds, et

de la gestion de la distribution des tâches sur les différents nœuds du réseau [4].dans ce qui suit en détaillera leurs fonctionnements.

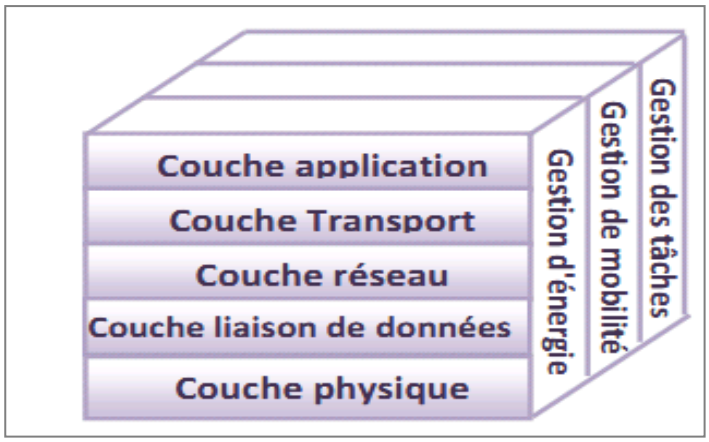


Figure I. 3: Pile protocolaire des réseaux de capteurs sans fil [4].

6.1. les couches protocolaires :

- **La couche physique :**

Permet de moduler les données et les acheminer dans le media physique tout en choisissant les bonnes fréquences.

- **La couche liaison de données :**

Spécifie comment les données sont expédiées entre deux nœuds/routeurs dans une distance d'un saut. Elle est responsable du multiplexage des données, du contrôle d'erreurs, de l'accès au media ainsi d'assurer la liaison point à point et multi-point dans un réseau de communication.

- **La couche réseau :**

Cette couche permet de trouver une route et une transmission fiable des données, captées, des nœuds capteurs vers le puits en optimisant l'utilisation de l'énergie des capteurs.

- **La couche transport :**

Cette couche est chargée du transport des données, de leur découpage en paquets, du contrôle de flux, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission.

- **La couche application :**

Cette couche assure l'interface avec les applications. IL s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels.

6.2. Les niveaux de gestion dans RCSF :

Les plans de gestions d'énergie, de mobilité et de tâche contrôlent l'énergie, le mouvement et la distribution de tâche au sein d'un nœud capteur. Ces plans aident les nœuds capteurs à coordonner la tâche de captage et minimiser la consommation d'énergie. Ils sont donc nécessaires pour que les nœuds capteurs puissent collaborer ensemble, acheminer les données dans un réseau mobile et partager les ressources entre eux en utilisant efficacement l'énergie disponible. Ainsi, le réseau peut prolonger sa durée de vie.

- **Plan de gestion d'énergie** :

  Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs. Dès lors, un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un nœud voisin afin d'éviter la réception des messages dupliqués. En outre, si le niveau d'énergie devient bas, le nœud diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage, conserver l'énergie restante au captage [8].

- **Plan de gestion de mobilité** :

  Détecte et enregistre le mouvement du nœud capteur. Ainsi, un retour arrière vers l'utilisateur est toujours maintenu et le nœud peut garder trace de ses nœuds voisins. En déterminant leurs voisins, les nœuds capteurs peuvent balancer l'utilisation de leur énergie et la réalisation de tâche;

- **Plan de gestion de tâche**:

  Lors d'une opération de capture dans une région donnée, les nœuds composant le réseau ne doivent pas obligatoirement travailler avec le même rythme. Cela dépend essentiellement de la nature du capteur, son niveau d'énergie et la région dans laquelle il a été déployé. Pour cela, le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau [8].

7. Conclusion:

Les réseaux de capteurs sans fil présentent un intérêt considérable et une nouvelle étape dans l'évolution des technologies de l'information et de la communication. Cette nouvelle technologie suscite un intérêt croissant vu la diversité de ces applications : santé, environnement et industrie.

Cependant, même s'ils apportent de nombreux avantages, les réseaux de capteurs posent un certain nombre de défis scientifiques. En effet, leurs limites en énergie, en capacité de stockage et de calcul ainsi que leur déploiement dans des environnements hostiles a fait que ces réseaux sont très vulnérables aux attaques. Le chapitre suivant sera consacré à l'étude de la sécurité dans les réseaux de capteurs sans fil.

Chapitre II : La sécurité dans les RCSF

1. Introduction:

Les RCSF connaissent actuellement une grande extension et une large utilisation dans différents types d'applications exigeant une grande sécurité. Nous avons abordé dans le premier chapitre les principales caractéristiques des RCSF, entre autres une densité importante, une capacité limitée de calcul et de stockage, une communication sans-fil et une ressource énergétique limitée.

En raison de ces contraintes ; les réseaux de capteurs doivent faire face à de nombreuses types d'attaques. Sans mesures de sécurité, un agent malveillant peut lancer plusieurs types d'attaques qui peuvent nuire au travail des réseaux de capteurs sans fil et empêcher leur bon objectif de déploiement. La sécurité est donc une dimension importante pour ces réseaux.

Ce chapitre traite les problèmes liés à la sécurité dans les RCSF. En commençant par l'étude des défis de la sécurité dans RCSF, puis les services de base de la sécurité et les mécanismes qui permettent de les assurer, et terminons par l'étude des menaces contre les RCSF et quelques contre-mesures pour les éviter.

2. Les vulnérabilités de sécurité dans les RCSF :

Le principal défi pour garantir la sécurité dans le RCSF découle paradoxalement de leurs caractéristiques qui les rendent si efficace et attractifs ; mais en même temps, constituent leurs handicapes et font que les mesures de sécurité doivent être obligatoirement restreintes, ci-dessous sont illustrés quelques exemples :

2.1. Limitation de ressources :

Chaque mécanisme de sécurité exige des ressources supplémentaires pour son exécution, ces ressources incluent la mémoire de données, ressources de calculs, et la source d'énergie pour actionner le nœud capteur ; cependant, ces ressources sont très limité dans chaque nœud capteur [6].

- **Limitations de mémoire et de la puissance de calcul :**

  Afin de mettre en application un mécanisme de sécurité efficace, l'algorithme utilisé pour une telle exécution doit avoir une petite empreinte de pas [16,15].

- **Limitations d'énergie** :

Un autre défi consiste à minimiser la consommation de l'énergie tout en maximisant les performances de sécurité. En effet, ces performances et les mécanismes de sécurité utilisés sont fortement influencés par les capacités et les contraintes du nœud capteur et l'énergie constitue l'essentiel des capacités du nœud capteur.

2.2. La communication non fiable :

La communication multi-sauts constitue une autre vulnérabilité des RCSF car la sécurité se fonde fortement sur un protocole défini, qui dépend de la communication [15,16].

- **Transferts incertains :**

Le développement d'un mécanisme de gestion d'erreurs pour éviter la corruption et l'abandon des paquets à cause des erreurs liés au canal lors de la communication ou de la congestion des nœuds**.**

- **Conflits :**

La mise en place d'un mécanisme qui prend en charge la gestion des conflits, permettant ainsi la minimisation des collisions et d'assurer une communication plus fiable.

- **Latence :**

Mise en œuvre d'un mécanisme de synchronisation pour résoudre les problèmes liés à celle-ci tel que la latence et qui peut être primordial pour les mécanismes de sécurité se basant sur les rapports d'erreurs et la distribution des clés cryptographiques.

2.3. Fonctionnement sans surviellance:

Les nœuds capteur sont souvent déployés dans des endroits hostiles. Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiquement éloignées ou inaccessibles [15,16]. Ci-dessous sont illustrés les principaux problèmes résultant du fonctionnement sans surveillance :

- **Exposition aux attaques physiques :**

Le déploiement des nœuds capteurs à proximité physique avec l'environnement conduit à de fréquentes compromissions intentionnelles ou accidentelles des nœuds. Comme les nœuds ne peuvent pas se permettre une protection physique inviolable, le potentiel de les attaquer et de récupérer leur contenu est important. Ainsi, les clefs cryptographiques et les informations sensibles devraient être gérées d'une manière qui augmente la résistance à la capture des nœuds.

- **Gestion à distance :**

Apres le déploiement d'un RCSF, la mise en place d'un mécanisme de gestion à distance pour la détection des attaques physique et les problèmes de maintenance du réseau est pratiquement impossible.

- **Pas d'administration centralisée :**

  Les réseaux de capteurs sont déployés sans point d'administration central. Cependant, si sa conception n'est pas adéquate, l'organisation du réseau serait difficile, inefficace et fragile.

3. Les objectifs de la sécurité :

Lorsqu'on aborde le problème de sécurité, on vise à atteindre certains objectifs, dont les principaux sont les suivants:

3.1. L'authentification :

C'est un mécanisme qui permet de vérifier l'identité des participants dans la communication, en distinguant les nœuds légitimes des intrus [13].En effet, lors de la communication entre deux nœuds légitimes n'importe quelle nœuds malveillant peut altère les paquets de données dans le but de les modifier ou bien d'injecter d'autre paquets supplémentaires [16]. Par conséquent**,** le récepteur doit s'assurer que les données utilisées proviennent d'une source légitime.

Donc l'authentification est nécessaire à la vérification de l'identité d'un émetteur de données et aussi à l'authentification de données transitant sur le réseau [20].

3.2. L'intégrité :

C'est un service qui permet de vérifier que les données ne subissent aucune altération ou destruction volontaire ou accidentelle [8], et conservent un format permettant leur utilisation lors de leurs traitements, de leurs conservations ou de leurs transmissions [15].

3.3. La confidentialité :

Une donnée ne doit être accessible qu'aux seuls nœuds autorisés. La confidentialité est la garantie que l'information d'un nœud n'est rendue accessible ou révélée qu'à son destinataire [8]. Ce service désigne la capacité du réseau à assurer ses services pour maintenir son bon fonctionnement en garantissant aux parties communicantes la présence et l'utilisation de l'information au moment souhaité. Comme les nœuds peuvent jouer le rôle de serveurs, la disponibilité reste difficile à assurer. En effet, un nœud peut ne pas servir des informations pour ne pas épuiser ses ressources d'énergie, de mémoire et de calcul en provoquant ainsi un mauvais comportement [20.6.13].

3.4. La disponibilité :

Ce service désigne la capacité du réseau à assurer ses services pour maintenir son bon fonctionnement en garantissant aux parties communicantes la présence et l'utilisation de l'information au moment souhaité [20]. Cette propriété reste difficile à assurer dans les RCSF étant données les contraintes qui pèsent sur ces réseaux [13,6]. En effet, un nœud peut ne pas

servir des informations pour ne pas épuiser ses ressources d'énergie, de mémoire et de calcul en provoquant ainsi un mauvais comportement.

3.5. Le contrôle d'accès :

Ce service consiste à empêcher des éléments externes d'accéder au réseau, et cela en attribuant aux participants légitimes des droits d'accès afin de discerner les messages provenant des sources internes du réseau de ceux externes [20.6.13]

3.6. Non-répudiation :

Ce service génère, maintient, rend disponible et valide un élément de preuve concernant un événement ou une action revendiquée de façon à résoudre des litiges sur la réalisation ou non de l'événement ou de l'action. C'est donc un mécanisme prévu pour assurer l'impossibilité que la source ou la destination puisse nier avoir reçu ou émis un message.

4. Les mécanismes de sécurité dans les RCSF :

Plusieurs mécanismes, basés généralement sur la notion de cryptographie, sont mis en place afin de répondre à la question de la sécurité dans les RCSF [10].

4.1. La cryptographie :

Elle est définie comme étant une science permettant de convertir des informations "en clair" en informations cryptées, c'est à dire non compréhensibles, et puis, à partir de ces informations cryptées, de restituer les informations originales.

Dans la cryptographie moderne, l'habilité de maintenir un message crypté secret, repose non pas sur les algorithmes, mais sur une information secrète dite clé qui est un paramètre utilisé en entrée d'une opération cryptographique et qui doit être utilisée avec les algorithmes pour produire le message crypté [14]. Il existe plusieurs outils cryptographiques dont nous citons **:**

4.1.1. Le chiffrement :

Le chiffrement est le système cryptographique assurant la confidentialité. Pour cela, il utilise des clés. [15] Selon cette utilisation, on distingue deux modes de chiffrement :

- **Le chiffrement symétrique :**

  Dans le chiffrement symétrique, une même clé est partagée entre l'émetteur et le récepteur. Elle est utilisée par l'émetteur pour chiffrer le message et par le récepteur pour le déchiffrer. [8,6]

- **Le chiffrement asymétrique :**

  Dans un système asymétrique, le récepteur génère une paire de clés asymétriques, une clé publique qui est diffusée à tous les émetteurs et une clé privée maintenue secrète

chez lui. Tout message chiffré avec la clé publique ne peut être déchiffré qu'avec la clé privée correspondante. [8 ,6]

Bien que la cryptographie à clé publique comporte des avantages certains par rapport à la cryptographie à clé symétrique, mais son utilisation dans les RCSF est très couteuse en termes de ressources, par conséquent la plupart des algorithmes proposés emploient les méthodes de chiffrement symétrique [10].

4.1.2. La signature digitale :

C'est un système cryptographique assurant la non-répudiation de la source. Elle repose sur les clés asymétriques. L'émetteur signe les données à transmettre avec sa clé privé (A) en produisant une signature digitale. Cette dernière est par la suite envoyée avec les données. Si elle peut être déchiffrée avec la clé publique (A) par le récepteur et si son résultat est identique aux données reçus alors la signature est valide, c'est-à-dire, les données proviennent de leur émetteur légitime qui ne pourra pas nier l'émission de ces données dans le futur [6].

4.1.3. La fonction de hachage :

C'est le mécanisme qui assure l'intégrité de données, il repose sur le principe qu'un message clair de longueur quelconque doit être transformé en un message de longueur fixe inférieur à celle de départ. Le message réduit nommé "hache" sera utilisé comme empreinte digitale du message original afin que ce dernier soit identifié de manière univoque [11].

4.1.4. Le code d'authentification des messages (MAC) :

Le code d'authentification de message MAC (Message Authentification Code) fait partie des fonctions de hachage à clé symétrique assurant l'intégrité de données comme toute autre fonction de hachage, en plus, l'authenticité de la source de données. [6]

Elle repose sur le principe que l'émetteur et le récepteur doivent partager une clé symétrique, qui sera utilisé par l'émetteur pour calculer un MAC sur le message à envoyer, ce MAC est la preuve d'authenticité qui accompagnera le message, ensuite le récepteur utilisera la même clé secrète pour calculer le MAC de nouveau sur le message reçus. Ainsi le MAC nouvellement calculé sera comparé au MAC accompagnant le message.

Si les deux valeurs sont égales alors le message et l'origine sont authentiques. Sinon l'un des deux n'est pas authentique. [8]

4.2. La gestion de clés :

La gestion de clés fournit des mécanismes efficaces, sécurisés et stables de gestion de clés utilisées dans les opérations cryptographiques. Par conséquence, c'est un service primordial pour la sécurité de n'importe quel système basé sur la communication.

La gestion des clés est l'un des aspects les plus difficiles de la configuration d'un système cryptographique de sécurité. Pour qu'un tel système fonctionne est soit sécurisé, chacun des utilisateurs doit disposer d'un ensemble de clés secrètes ou de paire de clés

publiques/privés. Cela implique de générer les clés et de les distribuer de manière sécurisée aux utilisateurs ou d'offrir à l'utilisateur le moyen de les générer. Il doit aussi pouvoir enregistrer et gérer ses clés publiques et privées de manière sure. [5,12]

La gestion de clés repose sur quatre fonctions pour assurer des services de sécurité tels que l'authentification, la sécurisation du routage et l'agrégation. Ces fonctions sont illustrées dans la figure ci-dessous :

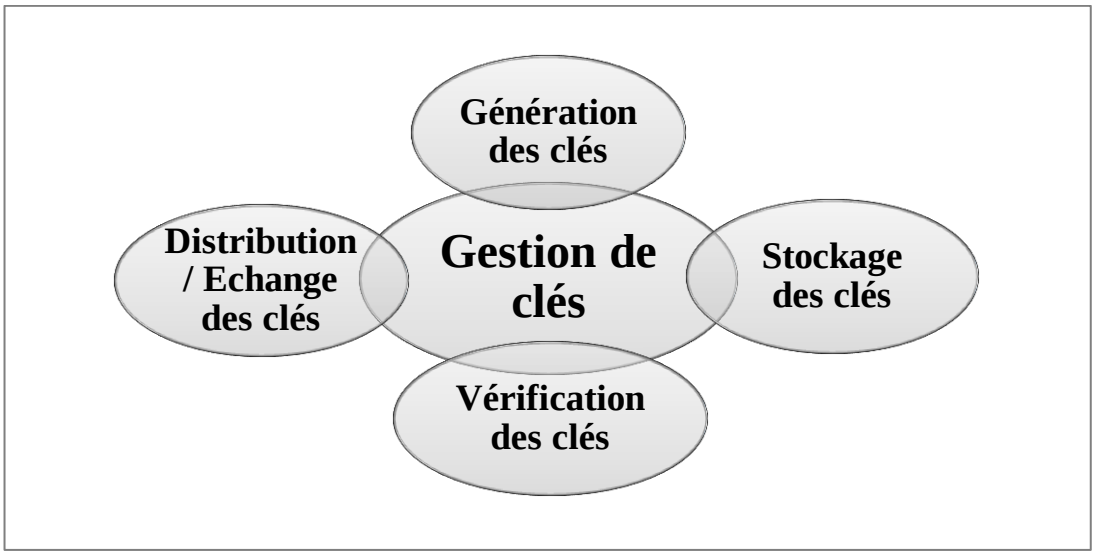


Figure II.01:Fonctions de la gestion de clés

La plupart des schémas de gestion de clés proposés pour les RCSF sont basés sur la cryptographie symétrique. Le problème majeur avec la cryptographie symétrique est de pouvoir trouver une méthode qui facilite l'établissement des clés entre les nœuds. La solution commune est d'utiliser une méthode de pré distribution dans laquelle les clés sont chargées dans les nœuds capteurs avant le déploiement.

Les solutions de gestion de clés basées sur la pré distribution sont classées dans plusieurs catégories selon la topologie du réseau (hiérarchique ou plate) et la façon avec laquelle les nœuds voisins partagent des clés communes (probabiliste ou déterministe).

Bien que la recherche sur des problèmes de sécurité dans les RCSF soit productive, le besoin d'un cadre de sécurité pour RCSF existe toujours.

Le système de détection d'intrusion (IDS) est un mécanisme commun de protection qui protège le réseau contre l'intrusion et les différents types d'attaques. Ce mécanisme serra développé dans le chapitre suivant.

5. Les attaques contre les RCSF:

Une attaque est un ensemble de techniques informatiques, visant à causer des dommages à un réseau, en exploitant les failles de celui-ci.

Les attaques connaissent plusieurs classifications envisageables dont les plus utilisées sont groupées selon les catégories ci-dessous :

5.1. La classification selon l'origine :

Dans la classification selon l'origine deux catégories sont distinguées : attaques internes et externe :

- **L'attaque externe:**

  Ce type d'attaque est déclenché par un nœud qui n'appartient pas au réseau, ou qui n'a pas la permission d'accès. L'objectif de cette attaque est de provoquer la congestion dans le réseau, de propager des informations de routage incorrectes, ou fermer complètement le réseau.

- **L'attaque interne:**

  Ce type d'attaque est déclenché par un nœud interne malveillant. Les stratégies de défense visent généralement à combattre les attaques externes. Cependant, les attaques internes sont les menaces les plus sévères qui peuvent perturber le bon fonctionnement des RCSF.

5.2. La classification selon la nature :

Du fait qu'un nœud malicieux peut opérer soit au niveau des données échangées entre les nœuds ou au niveau de la topologie du réseau, ses attaques peuvent être classées en deux catégories : actives et passives.

- **L'attaque passive :**

  Ce type d'attaque se limite à l'écoute et l'analyse du trafic échangé sans modifier les données ou perturber le fonctionnement du réseau. Donc, cette attaque est plus facile à réaliser mais difficile à détecter puisque l'attaquant n'apporte aucune modification sur les informations échangées .Une fois l'attaquant ayant acquis suffisamment d'informations, il peut produire un attentat contre le réseau, ce qui transforme l'attaque passive en une attaque active.

- **L'attaque active** :

  Ce type d'attaque vise la suppression ou la modification des messages transmis sur le réseau. Ainsi, injecter son propre trafic ou rejouer d'anciens messages en perturbant le bon fonctionnement du réseau.

6. L'affectation des attaques ou différentes couches dans les RCSF :

Une variété d'attaques contre les RCSF est rapportée dans la littérature. Pour faire face à ces attaques, diverses contre-mesures ont été proposées.

Nous présentons dans la suite les principaux types d'attaques, classifiées selon leurs affectations aux couches concernées de la pile protocolaire dans les RCSF.

6.1. Les attaques de la couche physique :

La couche physique est responsable de la spécification des caractéristiques matérielles, des fréquences porteuses, etc. Cette couche doit assurer des techniques d'émission, de réception et de modulation de données d'une manière robuste. Cependant, elle est vulnérable à certains types d'attaque parmi lesquels nous citons :

- **Brouillage (jamming) :**

  C'est une attaque qui vise les médias de communication utilisés dans les RCSF. L'attaquant peut émettre un signal d'une fréquence proche de celle utilisée dans le réseau afin de brouiller la communication. Cela empêche les nœuds d'échanger les données et provoque l'indisponibilité des canaux de transmission sans fil dans les RCSF [8].

- **L'attaque physique d'un nœud (tampering) :**

  C'est une attaque qui permet de reprogrammer, détruire ou violer un nœud légitime en accédant au logiciel ou aux matériels qu'il utilise [7]. Une fois que l'adversaire a pris le contrôle total d'un nœud légitime il essaye d'extraire des informations sensibles telles que les clés secrètes partagées entre les nœuds.

6.2. Les attaques de la couche liaison :

La couche liaison de données est responsable du multiplexage des flots de données, de la détection des trames, de l'accès ou médiums et le contrôles d'erreur. Les attaques visant la couche liaison incluent des collisions, épuisement, et injustice qui serrons détaillé ci-dessus.

- **L'attaque de collision :**

  Une collision se produit lorsque deux nœuds tentent de transmettre sur la même fréquence simultanément. Lorsqu'un paquet est en collision, un changement se produira probablement dans la partie des données provoquant un décalage de contrôle à la réception. L'adversaire envoie son signal dès qu'il écoute une transmission de la part des nœuds légitimes afin de causer une interférence. Le paquet serra alors rejeté comme étant invalide.

- **L'attaque d'épuisement (Exhaustion) :**

  Les collisions répétées peuvent être utilisés par un attaquant pour provoquer l'épuisement des ressources. Par exemple, le mécanisme naïf mis en œuvre au niveau de la couche liaison peut retransmettre les paquets corrompus. À moins que ces retransmissions sont découvertes ou empêché, l'énergie du nœud sera rapidement épuisée. [3]

- **L'attaque d'injustice (Unfairness):**

  Cette attaque peut se produire lorsqu'un attaquant essaie de dégrader les applications de temps réel en perturbant la transmission des trames ou cherche à abuser de la propriété coopérative de la couche MAC, cette attaque mène à la dégradation de la qualité de service.

6.3. Les attaques de la couche routage :

La simplicité de beaucoup de protocoles de routage pour RCSF à fait d'eux une cible facile pour les attaques. Ci-dessus sont présentés quelques exemples de ces attaques :

- **Le relais sélectif de paquets (Select forwarding)** :

Dans une attaque de transmission sélective, les nœuds malveillants ont pu empêcher d'expédier certains messages ou même les rejeter, par conséquent, ces messages ne propageraient pas par le réseau [16].

- **Le puits d'attaque (Sinkhole)** :

Dans cette attaque, le nœud malveillant se place généralement à un endroit stratégique et essaye d'attirer vers lui le plus de chemins possibles permettant le contrôle de la plus part des données circulant dans le réseau. Pour ce faire, l'attaquant doit apparaître aux autres comme étant très attractif, en présentant des routes optimales. Ainsi, l'attaquant peut supprimer tous les messages qu'il doit retransmettre ou permettre la mise en œuvre d'une autre attaque. Il convient de mentionner que les RCSF sont particulièrement vulnérables à cette classe d'attaques en raison de leur paradigme de communication, où tous les nœuds capteurs acheminent les données vers un seul nœud puits.

- **L'usurpation d'identité (Sybil):**

Dans cette attaque, les nœuds présentent des identités multiples aux autres nœuds du réseau, créant ainsi des inconsistances dans les tables de routages des nœuds voisins. Dans les algorithmes où la fiabilité du routage est implémentée par l'instauration d'une redondance de chemins, un attaquant peut altérer le fonctionnement du système en proposant plusieurs identités, ce qui permet de créer plusieurs routes passant par le nœud malicieux, qui ne sont en réalité qu'un seul chemin. [3]

- **L'attaque par inondation "HELLO" :**

De nombreux protocoles de routage utilisent des paquets "HELLO" pour découvrir les nœuds voisins et ainsi établir une topologie du réseau. La plus simple attaque pour un attaquant consiste à envoyer un flot de tels messages pour inonder le réseau et empêcher d'autres messages d'être échangés**.** [16.8]

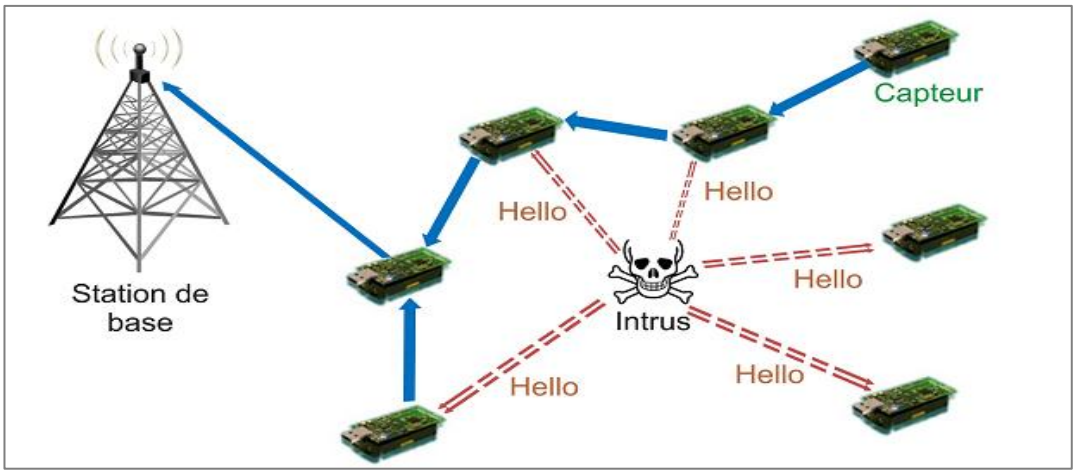


Figure II. 2 :l'attaque d'inondation.

- **L'attaque de trou ver (Wormholes):**

Connu aussi sous le vocable de tunneling, dans cette attaque, un adversaire peut recevoir des messages et les rejouer dans différentes parties à l'aide d'un tunnel entre les nœuds malicieux [8].

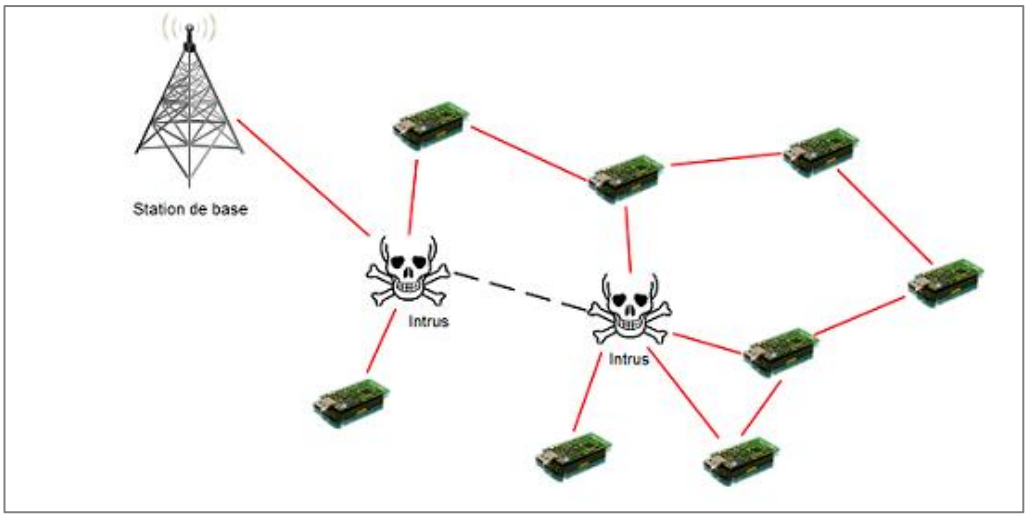


Figure II. 3:l'attaque de trou ver

- **L'attaque de trou noir (Blackhole Attack) :**

L'attaque du trou noir consiste tout d'abord à insérer un nœud malicieux par divers moyens dans le réseau. Ce nœud va modifier les tables de routage pour obliger le maximum de nœuds voisins à faire transiter leurs informations par lui. Ensuite, toutes les informations qui passent par ce nœud ne seront jamais retransmises, comme le montre la figure II.3.

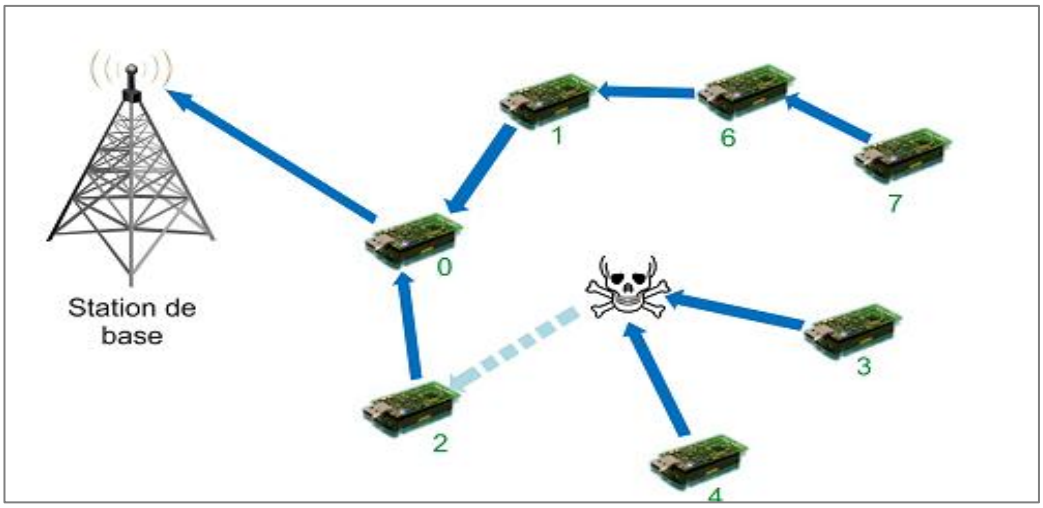


Figure II. 4: l'attaque de trou noir.

6.4. Les attaques de la couche application :

Cette couche est la plus proche des utilisateurs, géré directement par les logiciels, donc subit à certaine attaque quand présentera ci-dessus.

- **L'inondation (Flooding)** :

  Dans ce type d'attaques, le but de l'attaquant est d'épuiser les ressources du nœud victime (mémoire, énergie) en lui envoyant un grand nombre de requêtes successives d'établissement de connections [14].

- **Désynchronisation** :

  Consiste à rompre la connexion existante entre deux nœuds en resynchronisant leur transmission. L'attaquant emprunte une identité légitime à plusieurs reprise pour demander à un nœud la retransmission de paquets endommagés ou manquants, ainsi, si la synchronisation du nœud est correcte, cela peut dégrader sa capacité d'échanger des données avec succès et épuiser son énergie pour corriger des erreurs qui n'ont jamais existé.

Le tableau ci-dessous résume les attaques classifiées selon leurs affectations aux couches concernées de la pile protocolaire dans les RCSF.

couches	attaques
physique	Jamming
	Tampering
Liaison de données	Collision
	Exhaustion
	Unfairness
	Spoofed, altered or relayed routing information

réseau	Selective forwarding
	Sinkhole
	Sybil
	Wormholes
	Hello flood attacks
	Acknowledgment spoofing
transport	Flooding
	De-synchronization

Tableau. II.01 : Attaques contre les RCSF.

7. Conclusion:

La sécurité est très importante pour la réussite de plusieurs applications des réseaux de capteurs. Cependant assurer la sécurité de ces réseaux est une tâche très complexe en raison des ressources limitées des nœuds capteurs.

La pluparts des mécanismes développés pour assurer la sécurité dans les RCSF sont spécialisés pour des attaques précises et n'offrent généralement pas la possibilité de détecter de nouvelles attaques et de se défendre contre des nœuds internes compromis. D'où la nécessité d'utiliser des solutions basées sur la détection des intrusions dans les RCSF qui semblent offrir de meilleurs résultats [26].

Le chapitre suivant sera consacré à l'étude des systèmes de détection d'intrusions.

Chapitre III : Les systèmes de détection d'intrusion dans les RCSF

1. Introduction :

Les réseaux de capteurs sans fil sont particulièrement vulnérables à des attaques diverses à différentes couches de la pile protocolaire. Plusieurs mécanismes ont été proposés pour assurer la sécurité dans ces réseaux tels : la cryptographie symétrique, l'authentification et la sécurité des protocoles de routage [8,15].

La difficulté avec ces solutions est qu'elles sont pour la plupart hautement spécialisées pour une attaque précise et elles n'offrent pas la possibilité de détecter de nouvelles attaques. D'où la nécessite d'utiliser une solution permettant la détection des attaques inconnues ainsi que toute violation de ces politiques de sécurité : les systèmes de détection d'intrusion.

Dans ce chapitre nous présenterons les systèmes de détection d'intrusion, leurs propriétés, les techniques de détection d'intrusion, ainsi que leurs architecture et composants, et nous terminerons par quelques modèles de système de détection d'intrusion existants.

2. Système de détection d'intrusion :

Avant de définir un système de détection d'intrusion, il est commode de définir la notion d'intrusion : Une intrusion est définie comme étant un accès non autorisé, autrement dit est un ensemble d'actions d'internes ou externes du réseau qui violent des aspects de sécurité d'une ressource de réseau ou de système [24].

Un système de détection d'intrusion est un système qui essaie de détecter et alerter une tentative d'intrusion système ou réseau [17]. Il est défini aussi comme étant un ensemble d'actions qui découvrent, analysent et rapportent des activités non autorisées et préjudiciables.

Le but des IDS est de détecter des violations de confidentialité et d'intégrité, et la disponibilité réduite des ressources. Un IDS surveille le réseau et améliore l'activité de l'utilisateur pour détecter l'intrusion. [18].

3. Les composants d'un IDS :

Un système de détection d'intrusion se compose généralement des composants illustrés ci-dessous [19] [26] [27] :

- **Les agents locaux** :

Ils surveillent les activités locales d'un nœud capteur[27]. C'est-à-dire : ils vérifient l'information traitée par le nœud, les lectures de capteur ou des paquets transmis par le nœud. Ils sont imbriqués dans chaque nœud capteur et activés lorsque le nœud doit traiter de l'information.

- **Les agents globaux :**

Ils surveillent les communications de leurs voisins, traitant l'information qui peut être extraite. Ils sont également situés dans chaque nœud et exécutent leurs tests périodiquement. En outre, il est possible d'ajuster leurs niveaux d'activation dus à la redondance du réseau. Noter que, pour les réseaux ayant une architecture en one-hop clusters, il peut être possible d'inclure l'agent global de nœud seulement dans les clusters head.

- **Les agents de station de base :**

Analysent la circulation de l'information obtenue à partir des nœuds capteur. Puisque la station de base est riche en ressources, elle peut être en activité à tout moment, exécutant des algorithmes complexes pour découvrir toutes les variations qui peuvent mener à localiser une intrusion.

Chacun des agents défini précédemment devrait se composer des composants suivants [26] :

- **L'acquisition de données :**

Ce composant rassemble les données à partir des autres nœuds capteur. Ces données doivent être traitées et analysées ensuite stockées par le composant statistique [27]. La source de ces données dépend de types d'agents à partir desquels elles sont récupérées. Ainsi, pour les agents locaux les données sont obtenues depuis les paquets détectés et traités ou envoyés par le nœud. Pour les agents globaux les données sont obtenues depuis les paquets circulants sur son voisinage qui ne sont pas forcément destinés à lui.

- **Un composant statique :**

Son rôle est de stoker les données traitées et analysées par le composant d'acquisition. Le contenu principal de ce composant doit être partagé par les agents.

- **Un composant de détection :**

Il est responsable d'obtenir les résultats des composants définis précédemment puis analyser ces résultats pour signaler en cas de signes d'intrusions [27,19]. Dans les cas où le composant de détection considérerait qu'un nœud se comporte d'une manière anormale, il doit stocker cette information dans la base de données d'alerte afin d'économiser l'énergie, les tests liés au composant statistique doivent être exécutés périodiquement.

- **La base de données d'alerte** :

Elle est utilisée pour stocker les informations de sécurité générées par les agents. Le format et la taille de cette base dépend des protocoles utilisés. Cependant, elle doit contenir les informations suivantes : le temps de création, classification et source d'alerte [23,26].

- **Le composant de collaboration** :

Il peut être activé quand la communication avec d'autres parties du système ou du voisinage est nécessaire.

4. Les propriétés des IDSs dans les RCSF [22, 20,21] :

Un système de détection d'intrusion dans les réseaux de capteurs sans fil doit satisfaire les propriétés suivantes :

- **L'audit local :**

Un IDS, pour les réseaux de capteurs sans fil, doit fonctionner avec des données d'audit locales et partielles car dans ces réseaux, il n'y a pas de points centralisés (à part la station de base) qui peut collecter les données d'audit globales.

- **Les ressources minimales:**

Un IDS pour les réseaux de capteurs doit utiliser un nombre minimum de ressources. Les ressources physiques du réseau et des nœuds telles que la bande passante, l'énergie et la puissance de calcul sont limitées. En effet, la communication entre les nœuds pour la détection d'intrusion ne doit donc pas prendre toute la bande passante disponible.

- **Pas de nœud de confiance :**

Un IDS dans les réseaux de capteur ne doit faire confiance à aucun nœud car contrairement aux réseaux filaires les nœuds capteur peuvent être facilement compromis.

- **Distribué :**

C'est à dire que la collection et l'analyse de données doivent se faire dans plusieurs endroits (locations). De plus l'approche distribuée s'applique aussi pour l'exécution de l'algorithme de détection et la corrélation d'alertes.

- **Sécurisé :**

Un IDS doit être capable de résister aux attaques. La compromission possible de surveillance d'un nœud ne devrait pas avoir un impact négatif sur le fonctionnement normal des nœuds légitimes.

- **Disponibilité :**

Un IDS devrait fonctionner continuellement et rester transparent au système et aux utilisateurs.

- **Réponse automatisée :**

Un IDS devrait avoir une réponse appropriée. En d'autres termes, il devrait non seulement détecter mais également répondre aux intrusions détectées et cela sans intervention humaine.

- **Exactitude et interopérabilité :**

L'exactitude d'un IDS dans les RCSF devrait inter-opérer avec d'autres systèmes de détection d'intrusions pour détecter en collaboration des intrusions.

5. L'architecture des systèmes de détection d'intrusion dans RCSF :

Les architectures des IDS dans les réseaux de capteurs sans fils peuvent être classées en trois catégories :

5.1. L'architecture autonome :

Dans cette catégorie, chaque nœud fonctionne comme un IDS indépendant et il est responsable de la détection des attaques qui lui sont destinées. Ainsi, les IDS ne coopèrent pas et ne partagent aucune information.

L'avantage de cette architecture réside dans sa simplicité et le fait qu'un attaquant n'est pas capable de diffuser des informations erronées car chaque nœud est indépendant de l'autre.

5.2. L'architecture distribuée et coopérative :

Dans cette architecture chaque nœud exécute son propre IDS mais les IDS coopèrent afin de créer un mécanisme de détection d'intrusion global. Le problème majeur pour les IDS de cette catégorie est qu'ils causent une dégradation des performances du réseau par le trafic échangé entre les différents agents IDS, en plus du gaspillage de la bande passante limitée du réseau.

5.3. L'architecture hiérarchique:

Dans cette architecture, chaque nœud appartient à l'un des clusters qui sont distribués géographiquement à travers le réseau comme le montre la figure III.1. Dans chaque cluster, un nœud cluster head responsable du routage et de la détection des anomalies dans son cluster.

L'architecture hiérarchique minimise la surcharge du réseau puisque la coopération est réduite entre les clusters-head et leurs membres. Cependant, elle ne permet pas d'avoir une vision globale du réseau à cause de l'absence de coopération entre les différents clusters et reste par la suite inefficace contre certaines attaques distribuées.

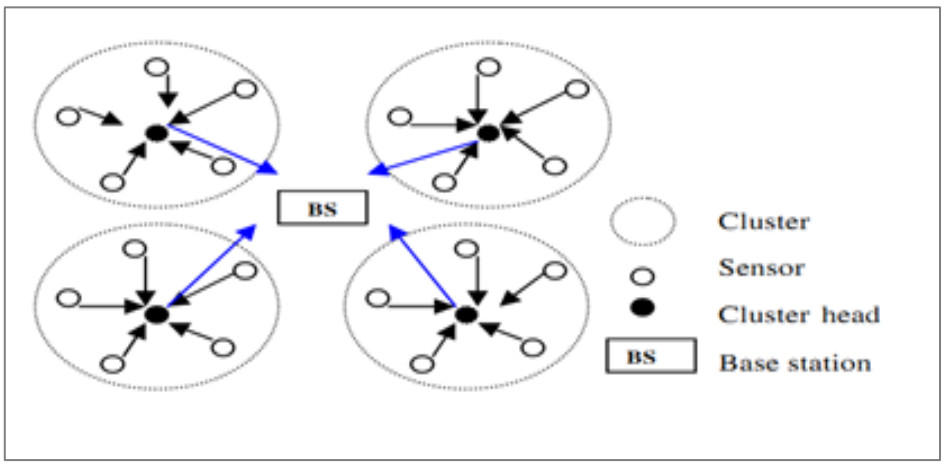


Figure III. 1 : l'architecture hiérarchique.

6. Les techniques de détection d'intrusion :

Afin de détecter un intrus, nous devons employer un modèle de détection d'intrusion. Un IDS doit pouvoir distinguer le comportement normal et anormal, afin de découvrir les tentatives malveillantes à temps.

La détection d'intrusion pour les réseaux de capteurs peut être classifiée dans trois larges catégories: détection par scénarios, détection comportementale et la détection par spécification.

6.1. La détection par scénarios :

Cette technique consiste à créer une base de signature d'attaques. En se basant sur celle-ci, une comparaison entre les comportements observés et les scénarios d'attaques prédéfinis dans cette dernière sera faite [18].Cette technique essaie de détecter l'évidence de l'activité intrusive indépendamment de n'importe quelle connaissance concernant le comportement normal du système.

L'avantage de cette technique réside dans l'efficacité de détecter les attaques connues dans la base de signature [19,25]. Par contre son inconvénient majeur est qu'on ne peut pas détecter des attaques originales [17,25]. La base de données de signature d'attaques doit continuellement être mise à jour de façon à ce que [19] :

- Les règles de mise à jour doivent être effectuées d'une façon fiable et sécurisée, elles ne sont pas faciles à réaliser dans WSN.
- Les données de la base de signatures seront stockées sur les nœuds cependant la capacité mémoire de ces derniers est réduite.

6.2. La détection par comportement :

Dans cette technique un modèle de référence de comportements normaux du système est créé [25]. Le comportement observé du système cible est comparé aux comportements normaux et espérés. Si le comportement du système est significativement différent du comportement normal ou attendu, on dit que le système cible présente des anomalies et fait l'objet d'une intrusion [17,19].

L'avantage majeur de cette technique est de pouvoir détecter de nouvelles attaques. Cependant, elle génère souvent de nombreuses fausses alertes car une déviation du comportement normal ne correspond pas toujours à l'occurrence d'une attaque, et le profil normal doit être périodiquement mis à jour et les déviations du profil normal calculé [19,18].

6.3. La détection par spécifications :

Cette technique est également basée sur les déviations de comportement normal afin de détecter des attaques, mais le comportement normal est spécifié manuellement comme un ensemble de contrainte du système [17]. De cette façon, les comportements légitimes invisibles ne causeront pas un taux élevé de fausses alertes, comme dans l'approche de détection d'anomalie. En outre, puisqu'il est basé sur des déviations des comportements légitimes, elle peut encore détecter des attaques inconnues.

7. Les modèles d'IDS existants pour RCSF :

La détection d'intrusion est un aspect important dans le domaine de sécurité des WSN, plusieurs études ont été présentées [33]. Dans cette section, nous présentons quelle que modèles existante sur la détection d'intrusion dans WSN :

7.1.1. Auto organisée et criticité stochastique d'apprentissage (Self-Organized Criticality & Stochastic Learning Based IDS : SOCSL):

Doumit et. Al [32] ont proposés un modèle de détection d'intrusion qui se base sur la technique de détection comportementale et l'utilisation du mécanisme centralisé pour la collection de données. Ce modèle consiste a stocké tous les comportements normaux des nœuds dans une base de connaissance, puis traiter tous les activités incompatible des nœuds en les comparant la base de connaissance [31,33].

Ce modèle a comme avantage la facilité qu'il procure pour l'ajout de nouveaux nœuds. Son principale inconvénient est le groupement des nœuds individuels plutôt que l'infrastructure en réseau de capteur ; aussi il ne contient aucune directive de donnée pour les attaques a les qu'elles auxquelles ils ne peuvent pas résister [32,33].

7.2. Les IDS décentralisés :

A.P. Silva et Al. [28, 29,34] présentent un mécanisme centralisé et distribué d'IDS basé sur la technique de détection par spécifications .Ce modèle est utilisé pour protéger les couches réseau et transport.

Le processus de détection de ce modèle se base sur les trois phases suivantes : l'acquisition de données, l'application de règles et la détection d'intrusions.

- **L'acquisition de données :** Le nœud surveille et écoute le trafic réseau de sa zone de couverture, pour rassemblés les informations des nœuds voisins, puis les stocke en mémoire.
- **L'application de règles :** Dans cette phase, le nœud qui surveille le réseau analyse les données rassemblées dans la phase précédente, le but est de vérifier s'il y a un comportement anormal en comparant les données rassemblées à l'ensemble de règles prédéfinies. S'il y a violation de ces règles alors un compteur de nombres d'échecs sera incrémenté.
- **La détection d'intrusion :** Si la valeur de compteur dépasse une valeur limitée une alerte sera produite.

7.3. L'algorithme de détection des attaques de routage(ADAR):

Juneja et Al. [33] ont proposé ce modèle pour détecter les attaques de routage dans les WSN. Cet algorithme se base sur le modèle de fourmis, qui sont divisées en deux catégories : fourmis en avant et fourmis en arrière [33].

- **Les fourmis en avant** :

  Elles sont générées par le nœud source pour le nœud destination recueillant des informations au sujet de l'état du réseau sur son chemin.

- **Une fourmi en arrière :**

  Sert à l'utilisation des informations collectées pour mettre à jour les tables de routages des nœuds sur son chemin et analyse ces dernières pour détecter l'attaque. Chaque nœud maintient une table de notation où il stocke les informations suivantes : l'énergie restante, la durée de vie de la fourmi, le rapport de paquet envoyé et livrés. La fourmi en arrière analyse cette table de notation, puis compare cette dernière aux valeurs d'un seuil prédéfinies pour déterminer la fiabilité du chemin.

8. Conclusion :

Le domaine des systèmes de détection d'intrusion pour les réseaux de capteurs a connu une grande évolution ces dernières années afin de répondre aux exigences technologiques et offrir un éventail de fonctionnalités capables de satisfaire les besoins de tous les types d'utilisateurs. Beaucoup de travaux ont été déjà menés pour RCSF, mais chacun pour répondre à un besoin spécifique d'une application, technologie ou d'une topologie [33, 29,32].

Cependant, les défis actuels dans les IDS est de proposer des solutions génériques capables de servir de base sécuritaire pour toute application dans les RCSF et la génération des faux positifs et des faux négatifs émis par le système.

Le chapitre suivant sera consacré à l'étude conceptuelle de notre travail qui porte sur un système de détection d'intrusion dans RCSF.

Chapitre IV : le Système de détection d'intrusion proposé

1. Introduction :

Les systèmes de détection d'intrusion sont des systèmes de sécurité conçu pour surveiller les données circulant dans les réseaux de capteur sans fil pour détecter les intrus et faire face non seulement aux attaques connus, mais aussi à celles inconnues .

Nous proposons un système de détection d'intrusion qui se base sur la technique hybride pour la détection d'intrusion, ainsi que sur l'architecture hiérarchique pour l'organisation des nœuds du réseau en cluster.

Dans ce chapitre nous avons cerné la démarche suivie pour la conception du système de détection d'intrusion proposé, en commençons par la description de son principe de base, ainsi que ces principaux objectifs, puis nous avons détaillé son fonctionnement et nous avons conclu par son déroulement général.

2. L'idée de base :

L'idée de base de notre travail est de développer un système de détection d'intrusion utilisant la technique de détection hybride qui regroupe la technique de détection comportementale et la détection de signature, ce qui lui permet d'atteindre un taux élevé de détection et un taux faible de fausse alerte. Afin de minimiser l'impact de ce système sur les ressources limitées du réseau nous avons profité des avantages offerts par l'architecture hiérarchique qui permet de l'organiser en cluster d'où la répartition des rôles entre les différents nœuds. Parmi ces avantages la réduction de la surcharge du réseau du fait que la coopération est restreinte au cluster, ce qui engendre un moindre coût en ressources de calcul et énergétique. Le système de détection d'intrusion développé (hybrid and hierarchical intrusion détection system : HHIDS) vise à atteindre certains objectifs dont nous citons :

- **Offrir un niveau de sécurité acceptable:**
  Prendre en charge les services de sécurité requis pour le réseau et minimiser le taux de fausses alertes.
- **Faire face aux attaques :**
  Pallier aux principales attaques qui peuvent cibler le réseau.
- **Maintenir les performances du réseau:**
  Ne pas dégrader les performances du réseau tel que l'énergie, la capacité de calcul et la mémoire.

3. L'architecture hiérarchique de HHIDS :

Comme illustré dans la section précédente, nous avons choisi d'utiliser l'architecture hiérarchique afin d'organiser les nœuds du réseau en clusters, de ce fait HHIDS opère principalement au niveau des clusters head et la station de base comme suite :

3.1. Au niveau des clusters head :

Dans HHIDS, un cluster head est responsable de son cluster, sa fonction principale est de surveiller le trafic réseau provenant des nœuds de son cluster. Dans ce qui suit nous déterminons son rôle au niveau de ce dernier :

- D'assurer l'intégrité, la fraicheur des paquets échangés ainsi que l'authentification de la source.
- Protéger le nœud des éventuels.
- Détecter les comportements anormaux des membres du cluster.
- L'envoi d'alertes à la station de base lors de la détection de comportement anormal ou d'attaques.

3.2. Au niveau de la station de base :

La station de base est responsable de :

- La prise de décision basée sur les messages d'alertes reçues à partir des clusters head.
- Toute décision prise concernant la suspension d'un nœud sera envoyée à tous les nœuds du réseau afin d'éviter le traitement des messages provenant d'un nœud marqué comme malveillant.

Afin d'avoir l'architecture hiérarchique, on n'a pas opté pour une méthode spécifique, mais on à profiter de l'architecture hiérarchique fournis par le protocole de routage LEACH (voir l'annexe A).

Par conséquent, HHIDS est intégré dans tous les nœuds du réseau, cependant, il n'est fonctionnel au niveau d'un nœud que lorsque ce dernier est élu comme cluster head.

4. L'architecture interne de HHIDS :

HHIDS est constitué de deux modules, le module de détection d'intrusion intervenant au niveau des clusters head et le module de prise de décision qui n'intervient qu'au niveau de la station de base.

4.1. Module de détection d'intrusion :

Le module de détection d'intrusion est responsable du filtrage de toute donnée reçue, de la détection des comportements anormaux et des attaques. L'organigramme ci-dessous illustre le fonctionnement interne de ce module ainsi que l'approche hybride utilisée:

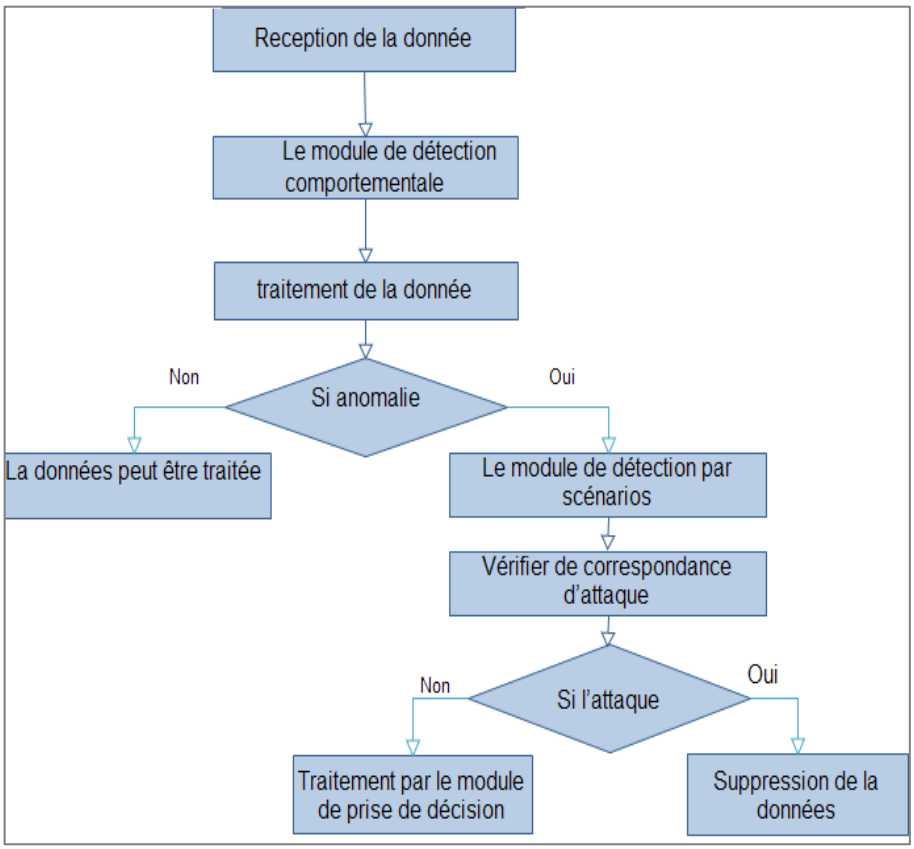


Figure IV.01: Le fonctionnement interne du module de détection d'intrusion.

4.2. Module de prise de décision :

Le module de prise de décision n'est sollicité que dans le cas où le module de détection d'intrusion ne parvient pas à reconnaitre l'attaque ou le comportement anormal et dans le cas où une attaque ou un comportement anormal est détecté à plusieurs reprises et provient d'une même source une alerte sera alors envoyée à la station de base qui exécute le module de prise de décision. Cette dernière envoie un paquet de décision à tous les nœuds du réseau afin de les mettre à jour par rapport aux attaquants et aux nœuds qui peuvent nuire le bon fonctionnement du réseau.

5. Les techniques de détection utilisées :

Comme illustré dans la Figure IV. 1, notre système de détection d'intrusion hybride qui regroupe les deux approches comportemental et par scénario. La première approche se base sur la définition des comportements normaux et toute déviation de ces comportements sera

considérée comme une intrusion. Le principal avantage de cette approche est de pouvoir détecter de nouvelles attaques.

Cependant, elle génère souvent de nombreuses fausses alertes car une déviation du comportement normal ne correspond pas toujours à l'occurrence d'une attaque. Cet inconvénient sera pallié grâce à la seconde approche qui consiste à faire une comparaison entre les comportements observés et les scénarios d'attaques prédéfinis. Le principal avantage d'une approche par scénario est la précision des diagnostics qu'elle fournit par rapport à ceux avancés par l'approche comportementale. Par contre son inconvénient majeur qui sera modéré grâce à la première approche est de ne pouvoir détecter que les attaques enregistrées dans la base de signatures.

Le choix de l'approche hybride a été motivé par le fait qu'elle permet d'avoir un taux élevé de détection et une meilleure exactitude. Ci-dessous nous illustrons les règles utilisés dans les deux modules : détection comportemental et détection par scénarios.

5.1. Règles utilisées pour la détection :

La différence entre une attaque et une faille système est que les failles système causent des erreurs aléatoirement, mais les attaques malveillantes sont faites délibérément et visent préférentiellement les composants les plus importants dans le réseau. En autre, les échecs peuvent exister partout dans le système et peuvent se produire à tout moment, mais la portée des attaques malveillantes dépend de la capacité des attaquants [48].

En termes de techniques disponibles, il y a des similitudes entre la détection des failles et la détection des attaques malveillantes. Puisque les erreurs provoquées par des failles et des attaques malveillantes sont des événements anormaux dans le système, il devrait être possible de détecter un tel événement en se rendant compte qu'il y a eu une déviation par rapport aux comportements normaux du système. Généralement, la technique de la détection d'anomalie peut également être employée pour détecter des anomalies autres que des failles et des attaques malveillantes [48].

Les règles ci-dessous sont utilisées pour la détection comportementale dans notre IDS [48] :

- **Règle d'intervalle :** Si l'intervalle de temps entre les réceptions de deux messages consécutifs est plus long ou plus court que les délais accordés, un compteur est alors incrémenté, dès que ce dernier atteint la limite permise une alerte sera envoyée à la station de base.
- **Règle d'intégrité :** Un message ne doit pas être modifié ou altéré durant sa transmission.
- **Règle de répétition :** Le même message peut être retransmis par un nœud seulement un nombre de fois limité.

- **Règle de blocage :** Le nombre de collisions liées à un message envoyé doit être inférieur au nombre prévu dans le réseau.
- **Gamme de transmission par radio :** Tous les messages reçus doivent être à partir d'un nœud membre du groupe.
- **Règle de fraicheur :** un message ne doit pas dépasser un délai déterminé entre son envoi et sa réception.

En surveillant régulièrement les violations des règles énumérées, les anomalies de réseau seront détectées.

6. Le schéma de sécurisation de HHIDS :

L'objectif principal de HHIDS est de protéger le réseau des éventuelles attaques, ainsi, son fonctionnement est concentré sur les données échangées entre les nœuds du réseau. Pour cela nous commençons par déterminer le format des paquets échangés au sein du réseau.

6.1. Format des paquets à envoyer :

Comme HHIDS intervient au niveau de deux couches, nous décrivons le format des paquets échangés au niveau de chaque couche :

6.1.1. Paquets de la couche réseau :

En plus des quatre types de paquet défini par le protocole LEACH, HHIDS utilise deux autres types, ci-dessous nous déterminons le format de tous ces paquets échangés ainsi que leur utilité :

- **Paquet de données :**

  Ce paquet englobe les informations envoyées d'un nœud à un cluster head ou d'un cluster head à la station de base.

- **Paquet de jointure :**

  Ce paquet est envoyé par les nœuds du cluster au cluster head pour l'informer de l'apparition à son cluster.

- **Paquet de Schedule :**

  Ce paquet est envoyé à tous les nœuds de sa zone pour les informer leur appartenance où réseau.

- **Paquet de Cluster head :**

  Ce paquet est généré par le nœud élu comme cluster head afin d'informer les autres de son nouveau statut.

- **Paquet d'alerte :**

Généré par le cluster head lorsqu'un nœud du cluster est la source de plusieurs attaques ou comportements anormaux afin de solliciter la station de base pour prendre une décision à propos du nœud malveillant.

- **Paquet de décision :**

Généré par la station de base pour informer les nœuds du réseau de la décision prise à propos d'un nœud marqué par l'un des clusters head.

6.2. Les services de HHIDS :

Dans ce qui suit nous déterminons la manière dont HHIDS assure les différentes règles définis précédemment :

6.2.1. Règle d'intervall**e** :

Si l'intervalle de temps entre les réceptions de deux messages consécutifs est plus long ou plus court que les délais accordés, un compteur est alors incrémente, dès que ce dernier atteint la limite permise une alerte sera déclenché.

6.2.2. Règle d'intégrité :

L'intégrité de données peut être assurée par le code d'authentification de messages MAC, cependant, l'application du MAC sur un paquet nous permet aussi d'assurer l'authentification. Pour cela, un nœud calcule le code MAC avant chaque envoie de paquets sur la données à envoyer et qui peut être l'un des paquets de la couche réseau définit précédemment et qui sont abrégé par le terme Data, notons que le champ MAC est initialisé à 0.

Le calcul du MAC sur l'identificateur du nœud permet d'assurer l'authenticité et pour assurer l'intégrité ce dernier doit être calculé sur l'identificateur du nœud ainsi que sur la données a envoyés comme suit :

$$\text{MAC (Data) tel que Data} = \text{L'Identifiant}_i + \text{les Données.}$$

Lors de la réception du paquet, le nœud recalcule la valeur MAC et la compare avec celle contenu dans le paquet, si ils sont égaux alors l'authenticité et l'intégrité des données sont assuré.

6.2.3. Gamme de transmission par radio :

Cette règle sera assurée par un cache d'identifiants des nœuds du cluster afin que le cluster head ne prenne en charge que les messages envoyés par les nœuds de son cluster. Ainsi, à la réception de chaque paquet l'identifiant de l'émetteur sera récupéré et comparé aux valeurs stockés dans le cache, et si aucun identifiant ne lui correspond alors le message sera

ignoré, cependant le nœud émetteur sera marqué et s'il renvoie d'autres messages jusqu'à atteindre un seuil prédéfini, une alerte sera alors envoyée à la station de base.

6.2.4. Règle de répétition :

Pour assurer cette règle, nous avons utilisé un cache de données qui contiendra tous les paquets reçu et pour chaque paquet reçu une variable entière sera associée afin de compter le nombre de fois que le même paquet a été reçu. Si cette variable dépasse un seuil prédéfini une alerte sera envoyée à la station de base.

6.2.5. Règle de blocage :

Cette règle est assurée au niveau de la couche liaison de données par le calcul du nombre de collisions causées par un même nœud. Une fois ce nombre dépasse un seuil prédéfini, une alerte sera envoyée à la station de base.

6.2.6. Règle de fraicheur :

La fraicheur des données est d'une importance à ne pas négliger puisque les données peuvent être rejouées par un adversaire. Pour différencier les données rejouées des données réelles, on utilise une empreinte temporelle. Pour assurer cette règle on rajoute à chaque paquet un champ TimeStamp. L'émetteur dépose dans ce champ la valeur du temps au moment de l'envoi.

Le nœud récepteur calcule la différence entre la valeur courante du temps et celle de TimeStamp. Si cette différence ne dépasse pas une certaine valeur, alors, le message n'a pas été rejoué.

7. Déroulement de HHIDS :

Dans cette section nous nous étalons sur le fonctionnement détaillé du système de détection d'intrusion HHIDS.

7.1. Au niveau du cluster head :

Les traitements effectuer lors de l'envoie et de la réception des paquets au niveau du cluster head sont les suivants :

- **A l'envoie :**

  A l'envoie, HHIDS effectue les traitements suivants sur le paquet à envoyer :

  - Calcule le MAC sur le paquet à envoyer et le met dans le champ MAC ;
  - Mettre le temps courant dans le champ TimeStamp, le temps courant est récupéré à l'aide de la fonction getTime() ;

- **A la réception :**

A la réception, HHIDS effectue les traitements suivants sur le paquet reçu :

- Il récupère la valeur du MAC envoyé dans le paquet dans une variable MAC_reçu.
- Il recalcule la valeur du MAC sur le paquet reçu.
- Compare la valeur du MAC_reçu et la valeur calcul. Si ces deux valeurs sont égales, alors le paquet du cluster head est authentique et intègre. Cependant il reste à vérifier si le paquet n'a pas été rejoué pour cette raison il faudrait avoir :

getTime ()-Packet.TimeStamp< £

(£ étant une certaine durée de temps prédéfinie).

- Effectue une vérification de l'identité du nœud émetteur si c'est un nœud du cluster, si oui le paquet sera alors soumis à la vérification suivante sinon le paquet ne sera pas traité.
- Vérifier si le paquet existe dans le cache de données, si c'est le cas alors le compteur associé à ce paquet sera incrémenté, lorsque ce dernier atteint un seuil prédéfini une alerte sera envoyée à la station de base
- Une fois ces tests sont effectués avec succès, on vérifie si le paquet existe dans le cache de données afin d'éviter l'attaque de rejoue et l'attaque d'épuisement, alors, si le paquet existe, on incrémente le compteur qui lui est associé et lorsque ce compteur atteint un seuil défini par l'administrateur une alerte sera envoyé à la station de base, dans le cas contraire c'est-à-dire le paquet n'existe pas dans le cache, le paquet sera enregistré et traité.

7.2. Au niveau de la station de base :

Les traitements effectués au niveau de la station de base lors de la réception d'un message d'alerte sont les suivants :

- La station de base effectue une vérification sur le code d'authentification de message MAC de la même manière que ce qui est décrit précédemment ;
- Si la vérification du MAC réussit, alors vérifier si le nœud cousant l'alerte appartient ou nœuds légitimes, si c'est le cas alors enregistre dans le cache d'alerte l'identifiant du ce dernier et dans le cas où une alerte concernant le même nœud est reçue une deuxième fois, la station de base diffuse un paquet de décision pour la suspension du nœud concerné ;
- si le nœud cousant l'alerte n'appartient pas aux nœuds légitimes, la station de base diffuse un paquet de décision pour la suspension du nœud concerné.

Lors de la réception d'un paquet de décision par les du réseau nœud, le nœud malveillant sera supprimé du réseau.

7.3. Détection de collision :

L'une des attaques qui peuvent survenir au niveau de la couche liaison de données est l'attaque de collision. Pour palier à ces attaques, toute trame défectueuse ne sera pas prise en charge. Cependant, cette trame sera soumise à certains traitements qui sont :

- Récupération de l'identifiant du nœud cousant la collision ;
- Incrémentation d'un compteur associé au nœud cousant la collision s'il est la source de plusieurs collisions ;
- Lorsque ce compteur atteint une limite prédéterminé, une alerte sera envoyée à la station de base ;

7.4. Détection de l'attaque de rejoue:

L'attaque de rejoue vise à épuiser les ressources en énergie et en capacité de calcul qui sont très limitées dans les réseaux de capteur sans fil. Pour se protéger de cette attaque et faire en sorte que le réseau aura la durée de vie prévue, les traitements suivants seront effectués sur toute trame reçue :

- Tout paquet reçu sera mis dans un cache de données et lui est associé un temporisateur et un compteur ;
- Si une même trame sera reçue durant le temps MinTime alors la trame ne sera pas traitée et le compteur sera incrémenté ;
- Si le compteur atteint une limite prédéterminée et avant que le temporisateur atteint MaxTime, alors une alerte sera envoyée à la station de base ;
- La trame est supprimée du cache une fois MaxTime est atteint.

8. Conclusion :

Le besoin de sécurité dans les RCSF est devenu un aspect important en raison de leur large utilisation dans plusieurs domaines. Cependant, il n'y a pas assez d'expérience sur la performance de la sécurité et la détection d'intrusions dans RCSF.

Dans le but de répondre aux exigences de sécurité dans la plupart des domaines d'application de ces réseaux, dans ce chapitre nous avons proposé un système de détection d'intrusion permettant de détecter certaines attaques et de détecter des comportements anormaux et d'y remédier et cela en intervenant au niveau de la couche liaison de données et la couche réseau.

Pour atteindre les objectifs de notre HHIDS , nous avons choisi les mécanismes de sécurité les plus adéquats qui prennent en charge les spécificités des RCSF notamment celles liées au ressources limitées en termes d'énergie, puissance de calcul et de mémoire.

Le prochain chapitre sera consacré à l'évaluation par simulation de la solution proposé.

Chapitre V :
Simulation et étude des résultats

1. Introduction :

Dans le chapitre précédent, nous avons proposé unsystème de détection d'intrusions pour sécuriser les RCSF. Dans ce chapitre, nous allons étudier ses performances à travers son implémentation. Cette étude consiste à simuler le comportement de l'algorithme proposé dans un réseau RCSF et de tester sa réponse contre quelques attaques.

L'objectif de ce chapitre est donc de démontrer l'efficacité du systèmes de détection d'intrusion proposé en termes de sécurité ainsi que d'autres métriques de performances. Pour cela, nous commencerons par définir les outils nécessaires pour l'implémentation et la simulation de notre système. Ensuite, nous décrirons la mise en œuvre de toutes les structures de données et processus décrits lors de la conception. Nous terminerons ce chapitre par une présentation des résultats relevés lors des tests de performances de notre système.

2. Présentation de J-Sim:

J-Sim [41, 42, 43] est un simulateur développé par une équipe du laboratoire DistributedRealtimeComputing Laboratoire (DRCL) de l'université d'Etat d'Ohio.Il est utilisé pour simuler le comportement des processus pseudo- parallèles, programmé en Java et il repose sur une structure logicielle basée sur des composants, appelée « Autonomous Component Architecture » (ACA). Le code source est organisé en paquetages relatifs à un type de composants.Un composant est une entité indépendante représentant un objet physique (une batterie, un module radio, une couche logicielle, etc.) ou logique (un protocole de routage, un modèle de mobilité, etc.). Ces composants seront ensuite connectés à l'aide de ports afin de générer un réseau simulé, et à afin de faire communiquer ces composant, des contrats sont imposer sur ces ports d'entrée/sortie.

La simulation du fonctionnement d'un réseau de capteurs, qui exige la définition des composants et leur mise en relation, est réalisée grâce à un langage spécifique, TCL (Tool Command Line) [44, 45, 46]. Il s'agit d'un langage de script dans lequel on spécifie l'architecture du réseau ainsi que les paramètres de simulation et d'analyse.

A l'aide de TCL, on définit les composants puis on les connecte. Tous les composants sont hébergés dans un conteneur, qui est à son tour un composant. La définition des composants est en fait la création des objets. Cette création est réalisée par la commande TCL mkdir.

Chaque composant est d'une entité indépendante qui fonctionne indépendamment des autres entités. Les composants possèdent des ports par défaut pour qu'ils puissent communiquer entre eux. D'autres ports peuvent être créés pour un composant.

Une connexion entre deux composants est réalisée par l'intermédiaire de deux ports dédiés, un dans chaque composant (voir la Figure V.1). Cette connexion est matérialisée par la commande TCL connect. Il suffit de suivre un schéma qui indique l'interconnexion entre les différents types de composants que l'on veut utiliser. On obtient ainsi l'architecture du nœud. Toujours dans ce script TCL, on y définit les paramètres globaux (par exemple la taille du champ de simulation), les outils de visualisation des résultats et l'ordonnancement de la simulation.

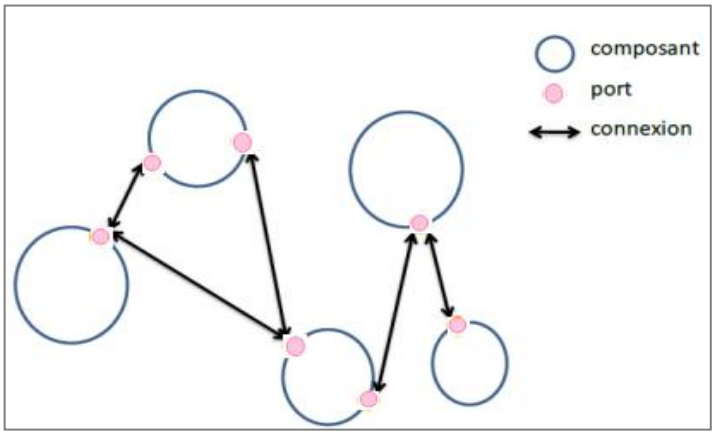


Figure V. 01:Connexions entre composants dans J-Sim

2.1. Types de nœuds :

Dans un réseaude capteurs sans fil, les capteurs surveillentun environnement spécifique(par exemple, un aéroport, un champ de bataille, etc.) et lors de la détectiond'unstimulus, les capteurs envoient des rapports à travers les nœuds sensor au nœud sink (la station de base).Il existe différents typesde stimuli, par exemple, acoustiques et sismiques, ces stimulisont généréspar des nœudstarget, par exemple, unréservoirmobilepeutgénérerdes vibrations du solqui peuvent être détectéspar des capteurssismiques. Ainsi, la simulation d'un environnement deréseaude capteurs sans filse compose de troisprincipaux types denœuds: sensor, target,et le nœud sink. Ci-dessous nous décrivons le mode de communication utilisé pour assurer la communication entre ces différents nœuds, puis nous déterminons le rôle de chaque nœud ainsi que son architecture interne.

Puisque la nature dela propagation du signalentre les nœudstargetet les nœuds sensor estfoncièrement différente de celleentre les nœuds sensor et le sink, deux modèles différentsde canaux de communication sont nécessaire : le canal sans fil "Wireless Channel", et le canal de capture "Sensor Channel".

- Le composant Wireless Channel est le composant commun pour tous les nœuds Sensor du réseau. Grâce à ce composant, la communication entre les nœuds sensor est possible ainsi qu'entre les nœuds sensor et le sink.
- Le composant Sensor Channel est responsable de la communication entre les nœuds target et les nœuds Sensor.

La figure ci-dessous montre les canaux de communication utilisés afin d'assurer la communication entre les différents nœuds du réseau.

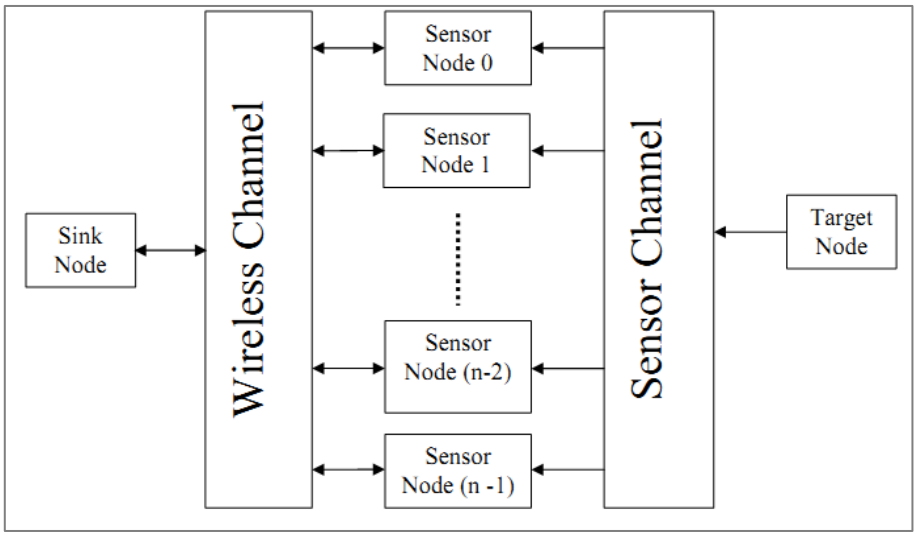


Figure V. 2: la communication des nœuds dans J-Sim

2.1.1. Les nœuds Target :

Les nœuds target ont pour fonction de générer des signaux (stimuli) qui peuvent être, par exemple, une secousse sismique, un bruit, etc. ce type de nœuds peut seulement envoyer des paquets de données sur le canal radio. La figure V.03 illustre l'architecture interne de ce nœud :

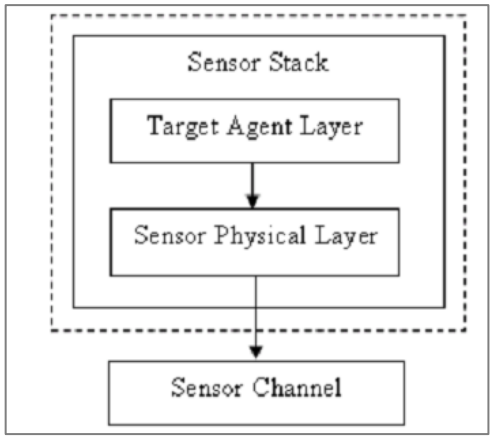


Figure V. 3: Vue interne d'un nœud Target

2.1.2. Le nœud Sink :

Le nœud sink est le nœud cible, celui où les informations doivent arriver. Il est doté d'une pile protocolaire lui permettant d'envoyer et de recevoir des données. La figure V.04 illustre les différentes couches constituant le module de communication de ce nœud :

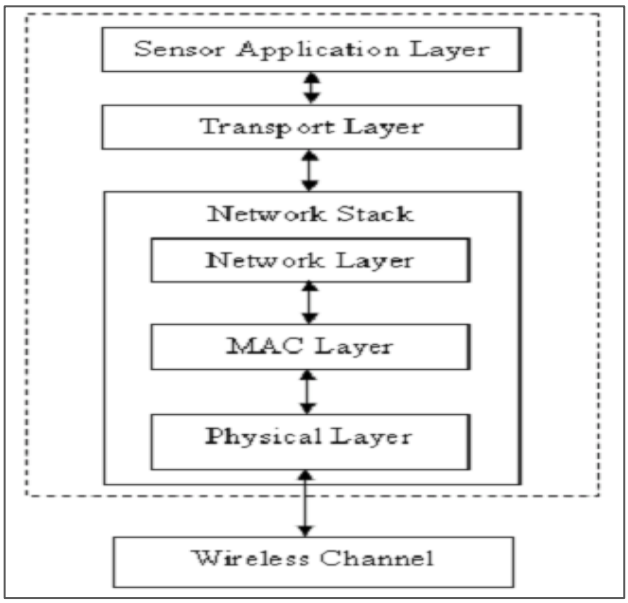


Figure V. 4: Vue interne d'un nœud Sink

2.1.3. Les nœuds Sensor :

Le reste du réseau est formé par les nœuds Sensor qui acheminent les paquets jusqu'au nœud Sink. Lorsqu'un nœud Target envoie un stimulus et qu'il est capté par un nœud Sensor, ce dernier crée un paquet contenant l'information de mesure pour la faire parvenir au nœud Sink via le réseau.

La Figure V.05 montre l'architecture interne d'un nœud Sensor dans J-Sim. On y distingue notamment les composants CPU et Radio qui définissent les modèles de la consommation énergétique du microcontrôleur et de l'antenne radio. Le composant CPU décrit en principe le coût de traitement des instructions. Le composant Radio décrit le coût de l'envoi et la réception des signaux et la mise en veille. Le composant Battery définit le modèle de la batterie d'un nœud. Ce composant calcule la consommation énergétique due au traitement des instructions, envois et réceptions des signaux en se basant sur les modèles du CPU et de Radio. Un programmeur peut facilement créer ses propres modèles (CPU, Radio, Battery) et les insérer dans un scénario de simulation afin de les tester. A noter que seuls les nœuds Sensor sont dotés d'une batterie.

La figure V.05 montre aussi que les composants des différentes couches d'un nœud sont séparés (Network layer, Mac Layer, Physical Layer). Ils communiquent entre eux via des ports. Les connexions (désignées par des flèches) entre les composants peuvent être unidirectionnelles ou bidirectionnelles.

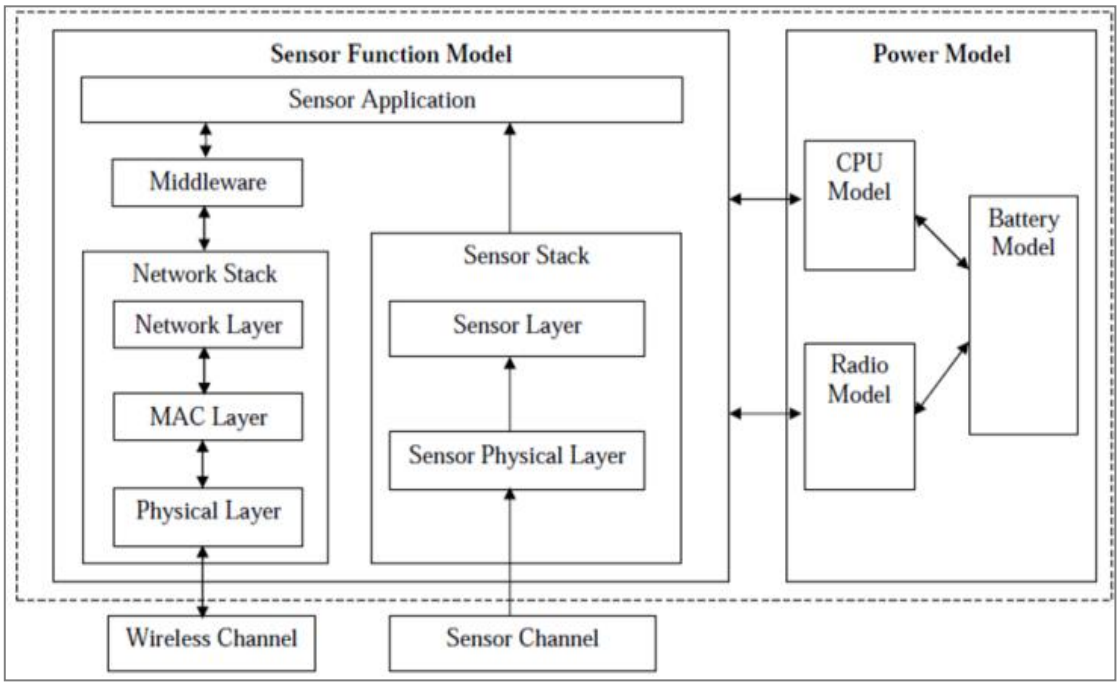


Figure V. 5:Architecture interne d'un nœud Sensor dans J-Sim.

3. Implémentation deHHIDS :

Dans cette section, nous allons présenter l'implémentation de la solution que nous avons discutée lors de la conception. Pour cela, nous commençons par la description des structures de paquets échangés entre les différents nœuds, puis nous nous étalons sur l'implémentation des algorithmes de détection utilisés et nous terminons cette section par la présentation de l'implémentation de l'attaque utilisée pour déterminer les failles de sécurité des RCSF sans présence d'un IDS et tester les performances du HHIDS.

3.1. La structure des paquets échangés :

Les structures utilisées dans HHIDS se résument au :

- **Paquet d'alerte :**

Cette structure est utilisée pour décrire un paquet d'alerte :

```
public LEACH_ALERT_Packet (long chId_,long nodeIdCauseAlerte_, int size_,int code_){
        this.chId=chId_;      //identifiant de cluster head
        this.nodeIdCauseAlerte=nodeIdCauseAlerte_;     //identifiant de nœud
                                qui a causé l'alerte
        this.code=code_;
        this.size = size_;
           }
```

- **Le paquet de décision :**

Cette structure est utilisée pour décrire un paquet de décision

```
public LEACH_Decision_Packet (long idSink_,long nodeSuspendu_,int size_,int code_){
        idSink=idSink_;  //identifiant de la station de base
        this.nodeSuspendu=nodeSuspendu_; //identifiant de nœud suspendu
        this.size=size_;
        this.code=code_;
           }
```

En plus de ces deux structures échangées au sein de HHIDS, des modifications ont été apportées aux paquets échangés. Ces modifications consistent en l'ajout de deux champs:

- TimeStamp : ce champ est utilisé pour vérifier que les données n'ont pas été rejouées.
- Mac : ce champ est utilisé afin de vérifier que les données n'ont pas été altérées durant la transmission.

3.2. Quelques fonctions de HHIDS :

Afin de contrôler le trafic réseau, HHIDS effectue des traitements sur toutes les données reçues. Dans cette section nous donnons l'implémentation de ses principaux traitements :

3.2.1. Le traitement de collision :

Comme appliqué dans la conception, toute collision est considérée comme étant un comportement anormal, lorsque cette dernière est causée par le même nœud à plusieurs reprises, le morceau du code suivant montre les traitements effectués sur toute collision détectée :

```
        protected synchronized void collision(Mac_Sensor_Packet p)
{
        ……..
   // vérifier si le cache des nœuds qui ont causé des collisiontabColl  n'est pas vide
if(j!=-1){
     // parcourirtabColl
 for (inti=0;i<j;i++){
// verifier une correspondence avec les noeuds enregistré dans le cache.
        if (tabColl[i].getNodeId()==pktRx_.getSa()){
            // incrémenter le compteur associer au noeud  dans le cas où le nœud  existe
            dans le // cache tabColl
        tabColl[i].nbre_Collision++;
        existe=true;
             // si nbCollisionCause>3 alors on envoie une alerte à la station de base
```

```
        if(tabColl[i].getNbreCollision()>3){
         // l'envoie d'alerte
        send(new LLPacket(0,this.getMacAddress(),12,new LEACH_ALERT_Packet
(pktRx_.getSa(),pktRx_.getDa(),12,-1)));
                    }// fin if
          } // fin if
        }  // fin for
      // enregistrer  le nœud dans le cache tabColl dans le cas où aucune
      correspondance n'a été trouver
if(!existe&& j!=0){
        tabColl[j]=new NodeCollisionCause(pktRx_.getSa(),1);
          j++;
        }// fin if
      }else{
        j=0;
        tabColl[j]=new NodeCollisionCause(pktRx_.getSa(),1);
          j++;
       }// fin else
……….
}
```

3.2.2. Traitement des paquets :

Le morceau de code suivant montre les traitements effectués sur les paquets de données :

a. A l'envoie d'un paquet :

```
protected void sendData(){
    …….
LEACH_Data_PacketnewPacket = new LEACH_Data_Packet (this.nid,
this.currentCH, phenomenon,
getTime(),  //recuperation du temps d'envoie ( = timeStamp)
this.code_, datasize, MAC.getMAC(newPacket1.toString())); // calculé du MAC sur
le message à envoyé
      …….
}
```

b. A la réception d'un paquet :

```
protected synchronized void recvDATA(LEACH_Data_Packetmsg){
boolean existe=false, // utiliser pour vérifier si le paquet existe dans le cache de
données
    traiter=false; // utiliser pour voir si le paquet nécessite un traitement ou non
int j=0;//utiliser pour parcourir  le cache de données
     //récupérer l'identifiant du nœud qui a envoyé la donnée
```

```
Long src_id = new Long(msg.getSender_id());
//récupérer  le code d'authentification de message
String mac=msg.getDATA_MAC();
// calculer la différence entre le temps courant et le moment d'envoi du paquet
double tem=getTime()- msg.getSendTime();
// verifier si le MAC calculé sur le message est égal au MAC contenu dans le message
      if(mac.equals(MD5.getMAC(msg.toString()))){
// verifiersi le paquet n'a pas été rejouer
if(tem < 1){
      // vérifier si le nœud appartient au membre du cluster
      if(joinedNodes.contains(src_id)){
      // vérifier que le paquet n'est pas envoyé par un nœud suspend du réseau
      If (!(cacheNodeSuspendus.contains(src_id ))){
      // vérifier si le paquet reçu existe dans le cache de données
      while(!existe&& j<cacheData.size()){
          if(j==0){
      cacheData.add(new CacheData(msg));
      traiter=true;
            j++;
           }else{
      LEACH_Data_Packetms=((CacheData)cacheData.get(j)).getPacket();
            if(msg.equals(ms)){
      inti=((CacheData)cacheData.get(j)).getNbfois();
                        ((CacheData)cacheData.get(j)).setNbfois(i++);
      existe=true;
                        if (((CacheData)cacheData.get(j)).getNbfois()>3){
      //l'envoied'alerte
      this.SendMyAlerteToBS(msg.getSender_id());
      compteurAlerte++;
      }    }
      j++;
       } }
       if (!existe){
      cacheData.add(new CacheData(msg));
      traiter=true;
            }
        ……..
      }
```

3.3. Implémentation des attaques :

Afin de vérifier le comportement de HHIDS dans le cas de présence de comportements anormaux dans le réseau, nous avons implémenté une attaque de rejoue qui vise principalement les clusters head qui représente les points les plus vulnérables dans le réseau.

```
protected void sendData(){
    …….
for (inti=0;i<10;i++){
   ……..
downPort.doSending(new
SensorAppWirelessAgentContract.Message(SensorAppWirelessAgentContract.BROADCAS
T_SENSOR_PACKET, this.currentCH, this.nid, datasize,LEACH_JOIN_REQ, eID, this.nid,
newPacket)) ;
    ……
}}
```

4. Métriques d'évaluation de HHIDS :

Dans cette section nous présentons l'ensemble des mesures utilisés pour tester le rendement de HHIDS :

- **Pertinence (Taux de Faux Positifs) :**

Cette notion concerne principalement le taux des faux positifs produits par un IDS dans un environnement donné pendant une période de temps particulière. Donc la fausse alarme est une alerte provoquée en absence d'attaque.

Dans notre cas, la détection de taux faux positif (FPR : False Positive Rate) est calculée avec la formule suivante [47] :

$$FPR = \frac{FP}{FP + TN}$$

FP (false positif) : c'est une fausse alerte au-dessus d'une activité légitime dans le réseau. Ceux-ci peuvent être produits d'adapter l'identification à un trafic non-malveillant normal.

TN (truenegative) : Ceci se produit quand il n'y a aucune activité malveillante ayant lieu dans le réseau. Par conséquent le TN peut être obtenu en soustrayant le TP du total le trafic surveillé.

$$TN = total\ paquets\ recus - TP$$

TP (truepositif) : Ceci de produit dans le cas de toute l'activité malveillante détectée.

- **Complétude (Taux de Faux Négatifs)** :

Cette notion concerne le taux d'attaques détectées correctement par un IDS dans un environnement donné pendant une période particulière. En parle aussi des faux négatifs, c'est-à-dire l'occurrence d'attaque en l'absence d'alerte.

Dans notre cas, la détection de taux faux positif (TPR : True Positive Rate) est calculée avec la formule suivante [47], TP la valeur définie précédemment :

$$TPR = \frac{TP}{TP + FN}$$

FN (false negative) : Ceci se produit quand HHIDS ne détecte pas une activité malveillante avoir lieu dans le réseau.

- **Consommation énergétique** :

Nous nous sommes intéressés essentiellement à la consommation d'énergie des nœuds puisqu'elle constitue un paramètre primordial pour la détermination de la durée de vie d'un RCSF. De ce fait, Nous analysons le surcout de détection HHIDS en termes de consommation d'énergie.

Pour ce faire, nous prenons comme critère, l'énergie restante dans l'ensemble des nœuds du réseau.

4.1. Paramétrage de la simulation

Avant de lancer les simulations, nous devons ajuster certains paramètres qui sont présentés dans le tableau V-1.

Les paramètres	La valeur
Le protocole de la couche MAC	TDMA, CSMA_CA
Le protocole de routage	LEACH
Nombre d'attaquants	[4,18]
Dimension du réseau	600*500
Energie associé à un nœud	0.25 joule
Nombre de round	5
Nombre total de nœuds	[50,150]

Tableau. V.01 :Paramètres du la simulation.

4.2. Résultats et interprétation :

Dans cette section, nous évaluons les métriques cités :

4.2.1. Pertinence et complétude :

L'histogramme ci-dessous illustre la complétude et la pertinence de HHIDS calculé à partir des résultats obtenu à partir de plusieurs simulation et cela pour les trois cas considéré : 50, 100, et 150 nœuds dans le réseau.

D'après l'histogramme en remarque que le taux de fausses alertes reçu par la station de base est insignifiant par rapport au nombre total d'alerte reçu, tout fois ce nombre est justifier par les collisions causé par les nœuds légitimes.

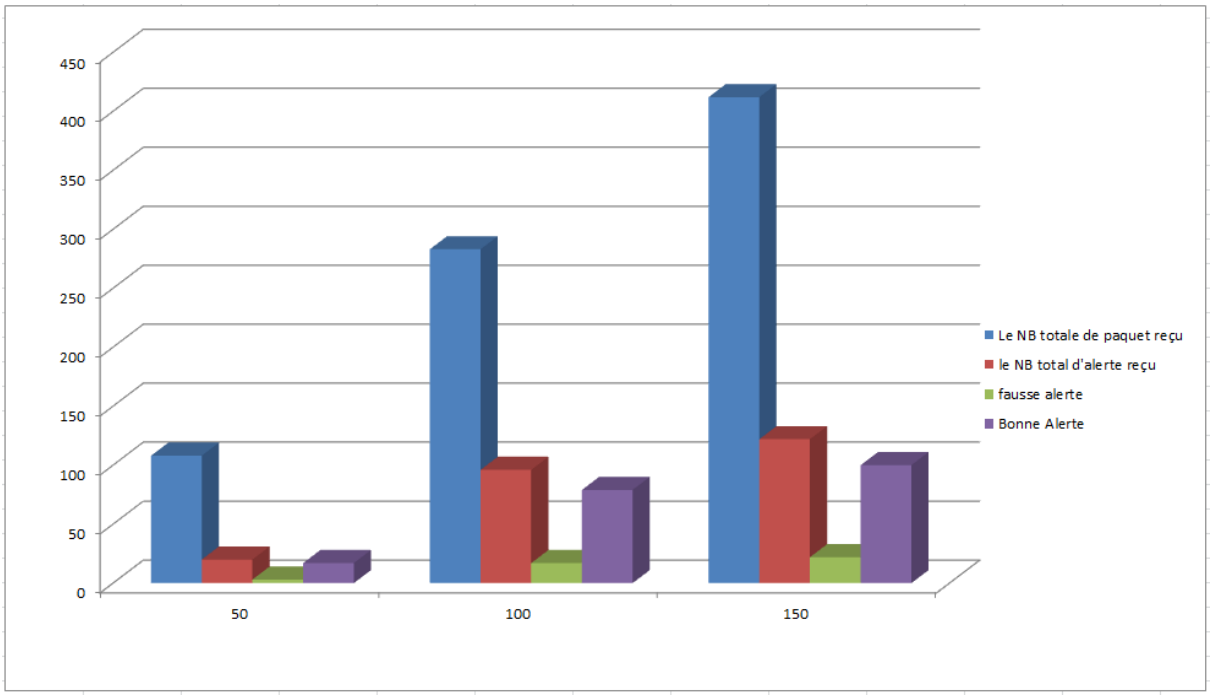


Figure V. 6:Histogramme représentant les paquets reçu par la station de base

TP : représente le nombre total d'alerte concernant les nœuds malveillant.

TN : Pour obtenir le vrai négatif on a calculé le nombre d'alerte reçu par la station de base en cas d'absence d'attaque dans le réseau.

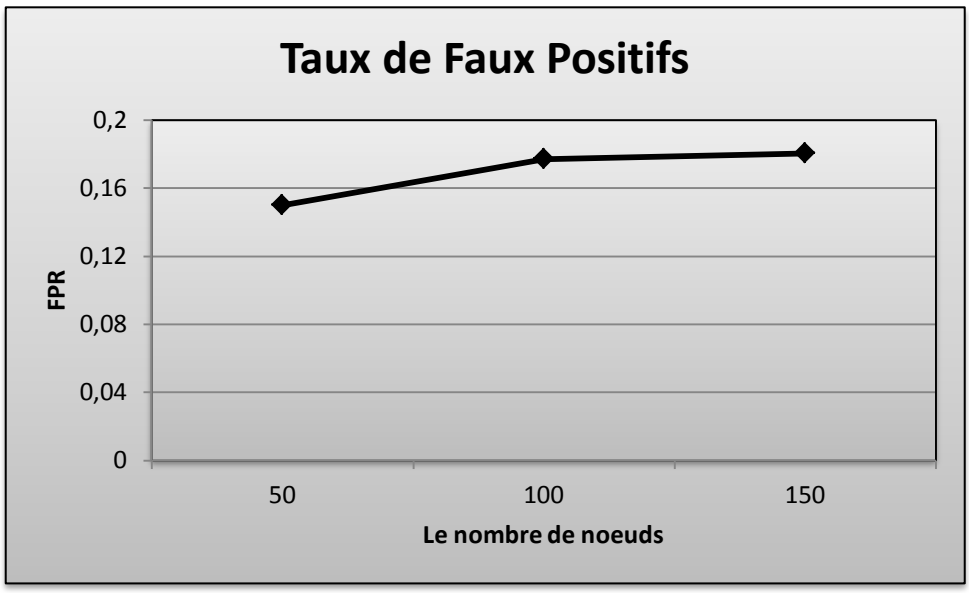
FP : Représente le nombre total d'alerte reçu concernant les nœuds légitimes.

FN : Représente le nombre de paquet des nœuds malveillant reçu et traiter par la station de base comme des paquets provenant de nœuds légitimes

A partir de ces valeurs récupérées, nous avons calculé la pertinence et la complétude de HHIDS en utilisant les formules citées précédemment.

Le graphe suivant représente le taux de faux positif produit par HHIDS, calculé en utilisant la formule présenté précédemment.

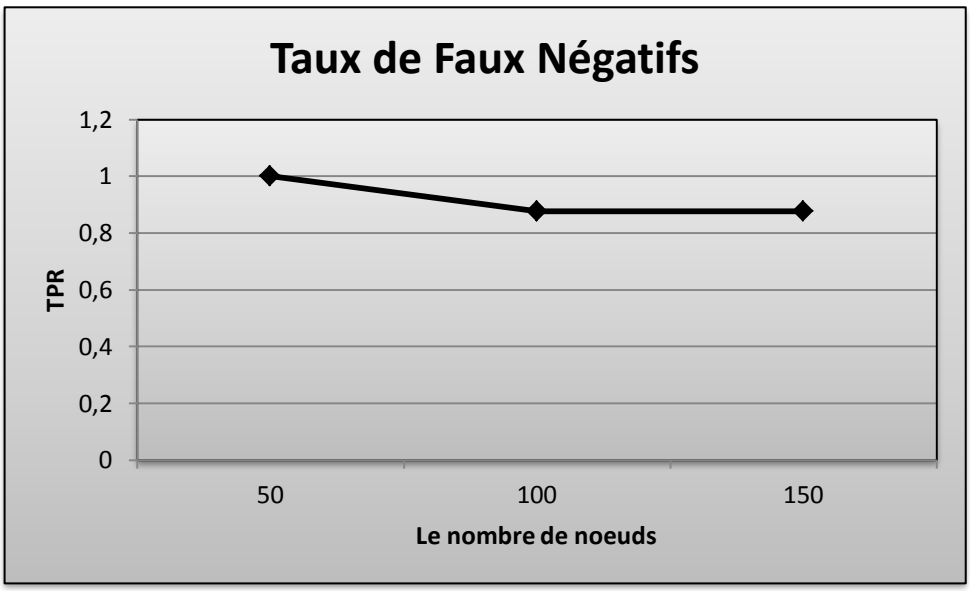
Le graphe illustre que le taux de faux positif tend vers 0.2 lorsque le nombre de nœuds augmente.



Gph.1 V-1: le taux de faux positif.

Le graphe suivant représente le taux de faux négatif produit par HHIDS, calculé en utilisant la formule présenté précédemment.

Le graphe illustre que le taux de faux négatif tend vers 0.8 lorsque le nombre de nœuds augmente.



Gph.1 V-2 : le taux de faux négatif.

En conclu des deux graphes cité précédemment que HHIDSproduit un taux de faux positif inferieures ou taux de faux négatif ,toute fois ce taux positif engendre un impact sur les ressources limités réseau, cependant ,du fait que ce dernier produit un taux de faux négatif

élevé on peut dire que l'objectif initial a été atteint mais la contrainte énergétique n'est pas vérifié.

4.2.2. Consommation d'énergie :

Apres avoir fait plusieurs simulations pour les mêmes paramètres avec et sans HHIDS, les résultats obtenu sur la durée de vie total de réseau sont similaire.

Ces résultats montre que même si HHIDS à un impact sur les ressources énergétiques, cela n'est pas vraiment pénible du fait qu'il permet de sécuriser les nœuds du réseau contre les attaques comme le montre la figure suivante où l'impact de l'attaquant apparais sur la durée de vie des nœuds voisins des attaquants.

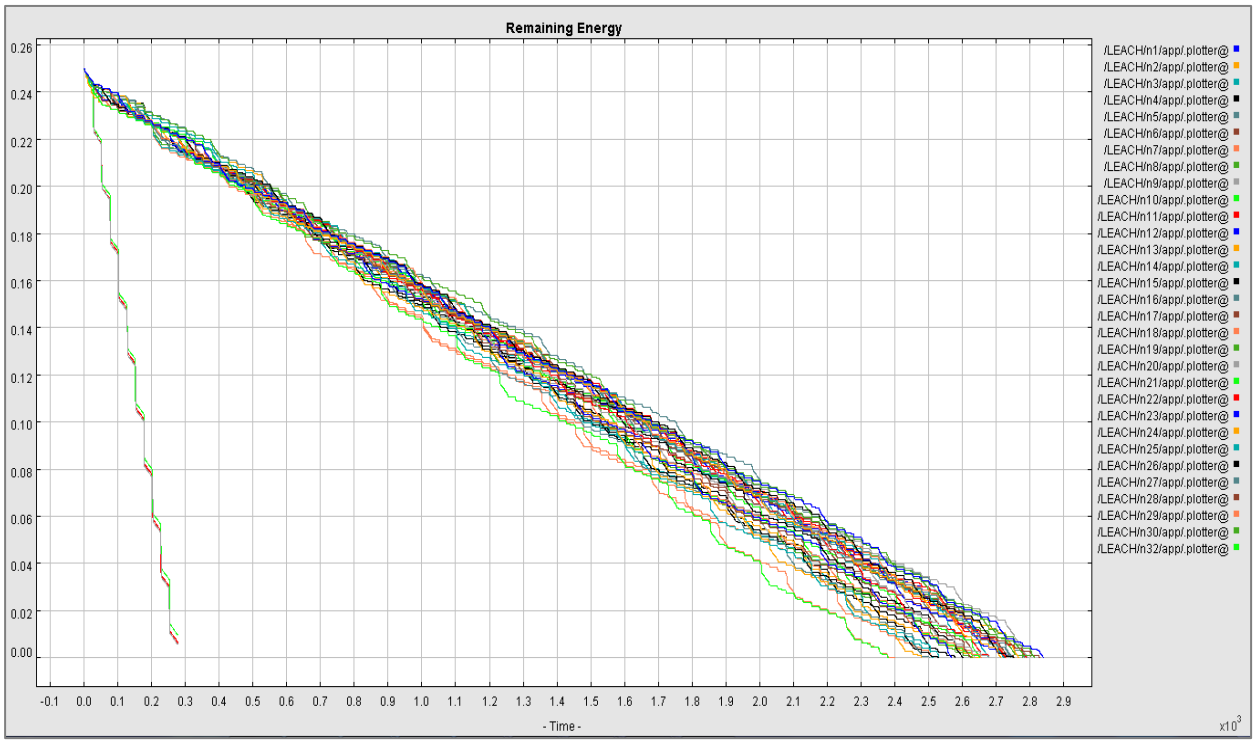


Figure V. 7:la durée de vie des nœuds sans HHIDS.

La figure suivante montre la durée de vie des différents nœuds du réseau en présence de HHIDS et des attaques dans le réseau, contrairement au cas précèdent la durée de vie de tous les nœuds à été maintenu durant toute la durée de vie du réseau.

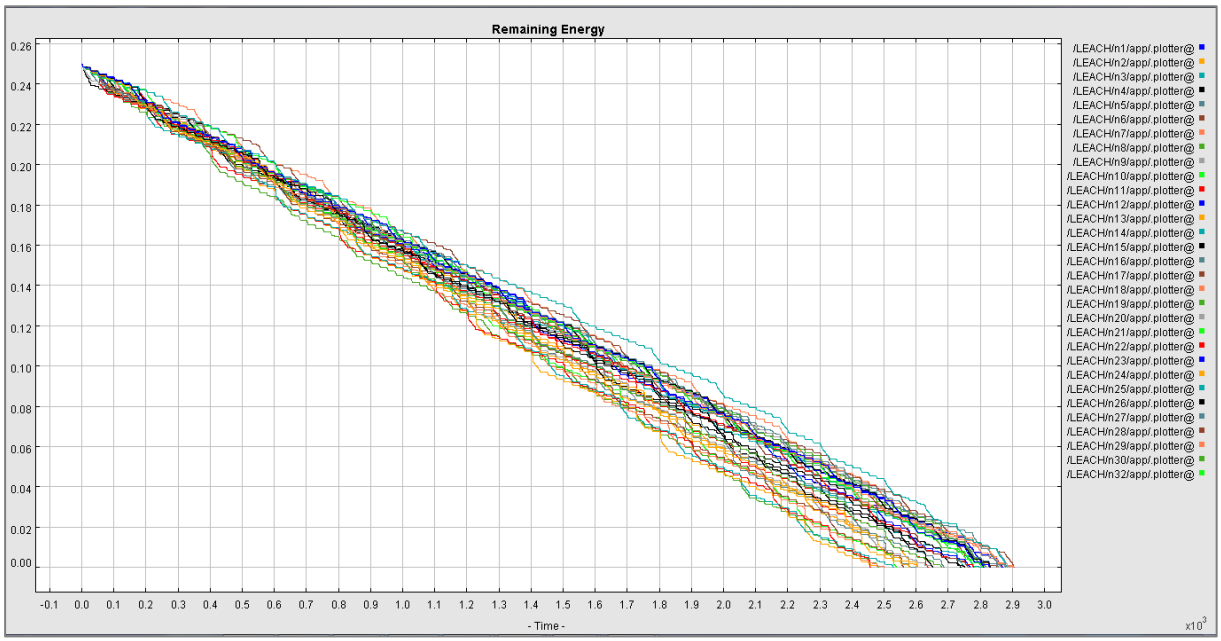


Figure V. 8: la durée de vie des nœuds avec HHIDS.

5. Conclusion :

La première partie de ce chapitre a été consacré à la présentation du simulateur utilisé, à la présentation des différentes structures de données utilisées par HHIDS et puis nous avons présenté l'implémentation des principaux traitements effectuées par HHIDS ainsi que l'implémentation de l'attaque qui vise l'épuisement des nœuds afin de voir les effets néfastes que l'absence de sécurité peut donner. En effet, une simple réinjection d'un paquet ou de données peut falsifier les résultats.

Dans la deuxième partie nous avons présenté les résultats obtenu lors de la simulation en prenant en compte certains métriques utiliser pour tester la performance d'un système de détection d'intrusion et nous l'avons terminer par la discussion des résultats. En effet, les simulations en montré que HHIDS produit un taux de fausses alerte qui a un impact sur les ressource du réseau.

Par ailleurs, nous avons constaté que les tests de performances effectués sur la sécurité des nœuds offerte par HHIDS, ont montré que ce dernier répond aux critères de performances souhaités.

Conclusion générale

Les réseaux de capteurs sans fil sont en plein développementet deviennent de plus en plus répandus. Actuellement, ils constituent un thème de recherche très dynamique vue leurs utilisations dans divers domaines. En effet, leurs applications sont de plus en plus nombreuses et diversifiées. Une problématique majeure dans ces réseaux est leur sécurité.

En effet, ces derniers sont très vulnérables à de multiples attaques vu leur contraintes critiques. Beaucoup de mécanismes ont été proposés pour assurer la sécurité dans ces derniers tels : la cryptographie symétrique, l'authentification et la sécurité des protocoles de routage. Cependant, la difficulté avec ces solutions est qu'elles sont pour la plupart hautement spécialisées pour une attaque précise et elles n'offrent pas la possibilité de détecter de nouvelles attaques. D'où la nécessite d'utiliser une solution permettant la détection des attaques inconnues ainsi que toute violation de ces politiques de sécurité : les systèmes de détection d'intrusions.

Dans ce travail nous avons proposé un système de détection d'intrusion qui se base sur la technique hybride pour la détection d'intrusion, ainsi que sur l'architecture hiérarchique pour l'organisation des nœuds de réseau (HHIDS). L'objectif de HHIDS est d'offrir un niveau de sécurité acceptable tout en optimisant les ressources du réseau et d'avoir un minimum de fausses alertes

Nous avons eu recours à la technique de détection hybride qui regroupe la technique de détection comportementale et la détection par scénarios, afin d'atteindre un taux élevé de détection et un taux faible de fausse alerte. Aussi de minimiser l'impact de ce système sur les ressources limites du réseau nous avons profité des avantages offerts par l'architecture hiérarchique qui nous a permet la répartition des rôles entre les différents nœuds, Dont les avantages peuvent se résumé a la réduction de la surcharge du réseau, ce qui engendre un moindre cout en ressource de calcule et énergétique.

Les résultats de simulation de HHIDS ont montré que ce dernier assure un taux de détection élevé, cependant le taux de fausses alertes produit par HHIDS est a assez considérable pour qu'il engendre un coût sur les ressources limitées du réseau, et montré

aussi que ce dernier assure la sécurité des nœuds du réseau et réduit l'impact d'un attaquant lorsque celui-ci est présenté dans le réseau.

Ceci dit, plusieurs perspectives sont envisagées pour notre projet tel l'utilisation d'algorithme de détection réduisant le taux de fausses alertes produite par HHIDS ainsi d'utiliser une politique à plusieurs niveau afin de minimiser les calculs et enfin envisager plusieurs couches de l'architecture protocolaire proposé par le modèle OSI et cela en prenant en considération la contrainte d'énergie qui un impact directe sur la durée de vie du réseau.

Références bibliographie

[1]Kaci BADER **« Détection d'intrusions dans les réseaux de capteurs sans fil »** Rapport de stage Master Recherche 2 en Informatique, IFSIC-Rennes 1, 2009/2010.

[2]LEHSAINI Mohamed **« Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique »** Thèse de Doctorat, Université A.B Tlemcen Faculté des Sciences pour l'Ingénieur && Université de Franche-Comté U.F.R Sciences et Techniques École Doctorale SPIM, Année 2009.

[3]ZIANE KHODJA Lilia**« La structuration et la sécurisation des réseaux de capteurs »**Master 2 Recherche Informatique, IFSIC.

[4]Khenfouci Yamina, Badaoui Amel **« Approche d'authentification dans les réseaux de capteurs pour la pédagogie »** Mémoire de fin d'études, Ecole nationale Supérieure d'Informatique (E.S.I, EX. INI) ,2008/2009.

[5] YACINE CHALLAL **«Réseaux de Capteurs Sans Fils »** 17/11/2008.

[6] Samir ATHMANI **« Protocole de sécurité Pour les Réseaux de capteurs Sans Fil»** Mémoire de Magistère en Informatique, Université Hadj Lakhder Batna, 15/07/2010.

[7] Gérard CHALHOUB **« Les réseaux de capteurs sans fil »** Clermont Université, IUT de Clermont-Ferrand, Dpt R&T, Complexe scientifique des Cézeaux, 63177 Aubière cedex, France.

[8]Messai Mohamed Lamine **« Sécurité dans les Réseaux de Capteurs Sans Fil »** Mémoire de Magistère en Informatique, Université Abderrahmane Mira de Bejaia, 2007 / 2008.

[9]Mr. Hoang DucChinh and Dr. Yen Kheng Tan **« Smart Wireless Sensor networks »** 2010 Published by InTechJanezaTrdine 9, 51000 Rijeka, Croatia, www.intechopen.com

[10]Jianying Zhou, Javier Lopez **« WIRELESS SENSOR NETWORK SECURITY »**Amsterdam • Berlin • Oxford • Tokyo • Washington, DC.

[11] C.H. Ngai « **Intrusion Detection for Wireless Sensor Networks »** The Chinese University of Hong Kong Department of Computer Science and Engineering, 2005.

[12] Noureddine LASLA **« La gestion de clés dans les réseaux de capteurs sans-fil »** Mémoire de Magistère en Informatique,Institut National de formation en Informatique (I.N.I) Oued-Smar, Alger ,2006 – 2007.

[13]Mi Chaw Mon THEIN ,Thandar THEIN **«A Framework for Secure and Survivable Wireless Sensor Networks »** The Annals of "Dunarea de Jos" University of Galati Fascicle I – 2009. Economics and Applied Informatics,Years XV - ISSN 1584-0409.

[14] YONG Wang, GARHAN Attebury, BYRAV Ramamurthy **«Survey of Security Issues in Wireless Sensor Networks» IEEE Communications SURVERYS,** 2ND QUARTER 2006, VOLUME 8, NO. 2 www.comsoc.org/pubs/surveys.

[15] BERRACHEDI Amel, DIARBAKIRLI Amina **« Sécurisation du protocole de routage hiérarchique LEACH dans les réseaux de capteurs sans fil »**Mémoire de fin d'études, Ecole nationale Supérieure d'Informatique (E.S.I) Oued-Smar, Alger, JUIN 2009.

[16] Luis E. Palafox, J. Antonio Garcia-Macias « **Chapter XXXIV Security in Wireless Sensor Networks »** Copyright © 2008, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

[17] Bc. LumírHonus**«Design, implementation and simulation of intrusion detection system for wireless sensor networks »**MASTER'S THESIS,MASARYKOVA UNIVERZITA FAKULTA INFORMATIKY, Brno, spring 2009.

[18] Md. Safiqul Islam, Razib Hayat Khan, Dewan Muhammad Bappy**« A Hierarchical Intrusion Detection System in Wireless Sensor Networks »**IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.8, August 2010.

[19] ANDRIY STETSKO**« INTRUSION DETECTION FOR WIRELESS SENSOR NETWORKS »**DISSERTATION THESIS TOPIC, FACULTY OF INFORMATICS MASARYK UNIVERSITY, Brno, 04.09.2008.

[20] AikateriniMitrokotsa , A. Karygiannis**« Intrusion Detection Techniques in Sensor Networks »**Wireless Sensor Network Security J. Lopez and J. Zhou (Eds.)IOS Press, 2008© 2008 The authors and IOS Press. All rights reserved.

[21] KrontirisIoannis,TassosDimitriou, Felix C. Freiling**« Towards Intrusion Detection in Wireless Sensor Networks »**.

[22] GiannetsosAthanasios **« Intrusion Detection in Wireless Sensor Networks »**MASTER THESIS, Carnegie Mellon University For the Degree of MASTER SCIENCE IN INFORMATION NETWORKING, April 8, 2008.

[23] Rodrigo Roman, Jianying Zhou,Javier Lopez**« Applying Intrusion Detection Systems to Wireless Sensor Networks ».**

[24] HosseinJadidoleslamy**« A High-Level Architecture for Intrusion Detection on**

Heterogeneous Wireless Sensor Networks: Hierarchical, Scalable and Dynamic Reconfigurable »Wireless Sensor Network, 2011, 3, 241-261.

[25] Michael riecker, Matthias hollic**« A SURVEY ON INTRUS ION DETECTION IN WIRELES S SENSOR NETWORKS»**Technical Report SEEMOO-TR-2011-02,

TechnischeUniversität Darmstadt Department of Computer Science Secure Mobile Networking Lab, www.seemoo.tu-darmstadt.de/dl/seemoo/seemoo-tr-2011-02.pdf.

[26]Bc. LukášFolkman**« Neighbour-based intrusion detection in wireless sensor networks »**MASTER THESIS, MASARYK UNIVERSITY FACULTY OF INFORMATICS, Brno, 2010.

[27] Rodrigo Romàn Castro**«APPLICATION-DRIVEN SECURITY IN WIRELESS**

SENSOR NETWORKS »submitted in fulfillment of the requirements for the degree of doctor in computer science at university of Malaga campus de teatinosblv. louispasteur, 35. 29071 MALAGA, 2008.

[28] Andreas A. Strikos, **« A full approach for Intrusion Detection in Wireless Sensor Networks»** , School of Information and Communication Technology KTH Stockholm, Sweden 16453,March 1, 2007

[29]AshfaqHussainFarooqi and FarrukhAslamKhan **« Intrusion Detection Systems for Wireless Sensor Networks: A Survey »**FAST National University of Computer and Emerging Sciences, 2009.

[30]Edith C.H. Ngai**«Intrusion Detection for Wireless Sensor Networks »**Ph.D. -- Term 2 Paper, The Chinese University of Hong Kong, 2005.

[31] Mohammad Saiful Islam Mamun , A.F.M. SultanulKabir **« 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.

[32]Harshal A. ArolkarShraddha P. ShethVaidehi P. Tamhane**«Ant Colony based Approach for Intrusion Detection on Cluster Heads in WSN »**India, February 12–14, 2011.

[33]Harshal A. Arolkar, Shraddha P. Sheth,Vaidehi P. Tamhane **« Comparative Analysis of IDS Algorithms for Wireless Sensor Networks »**Proceedings of the 5th National Conference; BharatiVidyapeeth's Institute of Computer Applications and Management, New Delhi , March 2011.

[34] Mohammed A. Abuhelaleh and Khaled M. Elleithy**«Security in Wireless Sensor Networks: Key Intrusion Detection Module in SOOAWSN »**University Of Bridgeport.

[35]

[36] Mahmood Ali and Sai Kumar Ravula**«REAL-TIME SUPPORT AND ENERGY EFFICIENCY IN WIRELESS SENSOR NETWORKS»** Master Thesis, HalmstadUniversity,Sweden, January 2008.

[37] HosseinJadidoleslamy**«A HIERARCHICAL INTRUSION DETECTION ARCHITECTURE FOR WIRELESS SENSOR NETWORKS»**International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.5, Sep 2011.

[38] Kamal BEYDOUN**« CONCEPTION D'UN PROTOCOLE DE ROUTAGE HIERARCHIQUE POUR LES RESEAUX DE CAPTEURS»** THESE, UNIVERSITE DE FRANCHE-COMTE, Année 2009.

[39] M. Yasser ROMDHANE **« Evaluation des performances des protocoles S-MAC et DirectedDiffusiondans les réseaux de capteurs»** Rapport De Projet De Fin d'Etudes, Ecole Supérieure des Communications Tunis, 2006 / 2007.

[40] Hung-Cuong LE **« Optimisation d'accès au médium et stockage de données distribuées dans les réseaux de capteurs »**THÈSE Pour l'obtention du Grade de Docteur,Université de Franche-Comté.

[41] Ahmed Sobeih, Jennifer C. Hou , **«A Simulation Framework for Sensor Networks in J-Sim »,**Department of Computer Science University of Illinois at Urbana-Champaign

[42]NGUYEN Phan Quang**, « Gestion de l'Accès aux Réseaux MPLS-DiffSer v par des Agents Intelligents »,** Mémoire de fin d'études du : DIPLOME D'ETUDES PROFESSIONNELLES APPROFONDIES, Institut de la Francophonie pour l'Informatique, Janvier, 2004

[43] Ahmed Sobeih, Wei-Peng Chen, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Ying Tyan, Honghai Zhang **« J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks»,**

[44] http://www.tcl.tk/man/tutorial/.

[45] http://ww7.be/neofutur/tools/tcl/.

[46] http://sites.google.com/site/jsimofficial/.

[47]OkoliAdaobi, Mona Ghassemian**«Analysis of an Anomaly-based Intrusion Detection System for Wireless Sensor Networks»** International Conference on Communication Engineering ,University Sistan and Baluchestan, December 2010.

[48]Qinghua Wang **«Traffic Analysis & Modeling in Wireless Sensor Networks and Their Applications on Network Optimization and Anomaly Detection »**Network Protocols and Algorithms, ISSN 1943-3581, 2010, Vol. 2, No. 1.

[49]Chong Eik Loo1 Mun Yong Ng,ChristopherLeckie, MarimuthuPalaniswami **«Intrusion Detection for Routing Attacks in Sensor Networks »**NICTA Victoria Laboratory,The University of Melbourne,

Annexe A : LEACH

1. Description du protocole LEACH:

LEACH (LowEnergy Adaptive Clustering Hierarchy)[2,36, 37, 38,39] est l'un des protocoles hiérarchiques les plus populaires. Ce protocole suit le modèle « Time-Driven » et utilise un clustering distribué (la formation des clusters et l'élection des cluster-heads se font au niveau des nœuds). LEACH suppose que les nœuds sont homogènes et que l'acheminement des paquets vers la station de base se fait en un seul saut via les cluster-heads. Les nœuds ont la possibilité de devenir cluster-heads en se basant sur des probabilités d'élection. LEACH s'exécute en cycles « rounds », comportant chacun deux phases, néanmoins, la phase d'initialisation et la phase de transmission.

1.1. La phase d'initialisation :

Cette phase est composée de trois sous- phases: d'annonce, d'organisation des groupes et enfin d'ordonnancement, et qui seront détaillée ci-dessous.

1.1.1. Phase des annonces :

C'est la phase où les clusters sont formés et les cluster-heads sont élus pour une période déterminée « round ». Cette élection se fait selon un algorithme spécifique d'élection qui prend en compte des différents critères comme l'énergie disponible dans les nœuds ou si le nœud a servi comme cluster-head pendant la dernière période. Tous les nœuds décident s'ils peuvent devenir un cluster-head indépendamment au même temps, et informent les autres nœuds quelque soient leurs décisions. Dès que les cluster-heads sont élus, chacun d'eux envoie un message de notification aux autres nœuds du réseau. Ces nœuds décident donc de leur appartenance à un cluster selon l'amplitude du signal reçu en choisissant le signal le plus fort.

1.1.2. Phase d'organisation des groupes :

Une fois la décision prise, chaque nœud doit informer son cluster-head de son choix par l'envoi d'un paquet d'affiliation et les clusters seront ainsi formés.

1.1.3. Phase d'ordonnancement :

Chaque cluster-head ayant reçu les messages des nœuds désirant appartenir à son cluster, diffuse un ordonnancement TDMA aux membres de son cluster en attribuant à chaque membre un intervalle de temps (TimeSlots) durant lequel il pourra communiquer ses données.

1.2. Phase de transmission de données :

Après l'établissement de l'ordonnancement, les membres communiquent et transmettent les données vers leurs cluster-heads (CHs) en un seul saut durant les slots qui leur ont été consacrés. Les CHs agrègent les données reçues et les transmettent vers la station de base. Le réseau réitère ensuite ces phases précédentes dans de nouveaux rounds. La figure ci-dessous illustre la répartition du temps et les différentes phases pour chaque round.

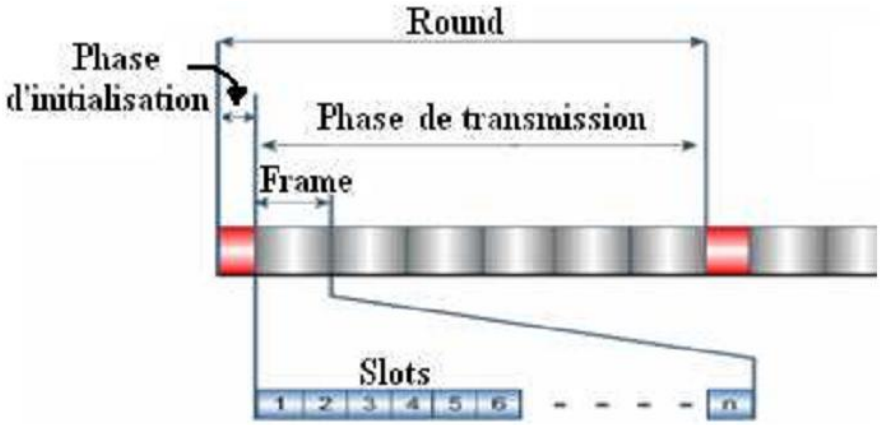


Figure A. 1: Répartition du temps et différentes phases pour chaque round.

2. Description des protocoles MAC utilisé par LEACH :

Durant son fonctionnement, le protocole LEACH appelle certains schémas des protocoles MAC, ci- dessous nous nous étalons sur la description du schéma d'accès multiple à répartition de temps (TDMA) sollicité par LEACH pour la répartition des TimeSlots entre les différents membres dans un cluster. Ainsi que le schéma d'accès multiple avec surveillance de porteuse (CSMA) sollicité lors de l'envoie de décision des cluster-heads et l'échanges de données [36,40].

2.1. Description de TDMA:

Le schéma d'accès multiple à répartition de temps ou TDMA (Time Division Multiple Access) permet de diviser le temps en intervalles (time-slot) attribués à chaque nœud ci-dessous. Ainsi, un seul nœud a le droit d'accès au canal (il utilise toute la plage de la bande passante du canal), mais doit émettre ses données pendant les intervalles de temps qui lui sont accordés.

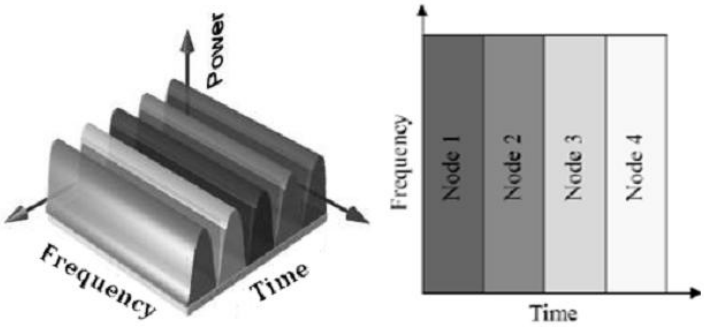


Figure A. 2: Diagrammes représentant le protocole MAC TDMA.

2.2. Description de CSMA :

Le schéma d'accès multiple avec surveillance de porteuse ou CSMA (Carrier Sense Multiple Access) est une technique dans laquelle chaque nœud doit rester à l'écoute du médium avant de transmettre ses données. Lorsqu'un nœud veut transmettre un message, il examine le média pour vérifier s'il est libre ou occupé par un autre nœud. Dans le cas où le media est libre, ce nœud pourra émettre son message afin d'éviter les collisions. Cela dit, des nœuds peuvent émettre des données en même temps, ce qui mène à des collisions. Il est nécessaire donc que celles-ci soient détectées et que la récupération de données soit effectuée et que ces données soient retransmises.

Si les retransmissions se passent encore en même temps, d'autres collisions vont se produire. Une solution à ce problème consiste à introduire que chaque nœud attende un délai aléatoire avant de retransmettre ses données, ce qui réduit la probabilité d'une autre collision.

Annexe B

1. Introduction :

J-Sim est un environnement de simulation utilisé pour évaluer les protocoles de routages proposés pour les réseaux et plus particulièrement ceux développer pour les réseaux de capteur sans fil. Dans cette annexe nous présentant les étapes à suivre pour installer et lancer J-Sim sous la plate forme Windows.

2. Préparation d'environnement :

- **Etape1 :**

Pour installer le simulateur J-Sim, il faut déjà avoir installé sur votre PC un logiciel spécial depuis le site web de la société Sun Microsystems, son nom complet est Java 2 Software Development Kit (J2SDK). Pour des raisons de stabilité, il est hautement recommandé d'installer la version J2SDK 1.4 ou bien des versions ultérieures. La version que j'ai utilisé pendant mon projet de est J2SDK 5.0 et peut être téléchargée depuis le site suivant : http://java.sun.com/j2se.

- **Etape2 :**

La compilation des codes sources pour le simulateur J-Sim peut se faire soit via le compilateur Apache Ant, soit par la commande « make ».Le premier compilateur est valable depuis le lien : http://ant.apache.org/ . Pour compiler via la commande « make », on doit installer l'utilitaire GNU-Makemakefiles in Java. Cet utilitaire permet d'installer des commandes propres aux plateformes Linux sous Windows.

- **Etape3 :**

La version du simulateur J-Sim la plus utilisé est J-Sim version1.3. Elle peut être téléchargée depuis le lien : http://www.j-sim.org/cgi-bin/j-sim_downloader.cgi. Il faut noter que la version J-Sim version1.3 n'est pas compatible avec la version J2SDK 5.0.

Pour des questions d'interopérabilités, on doit installer le patch4 ou bien des versions de patch ultérieures. Le patch dont j'ai installé est **patch4-version1.3** valable sur: http://www.jsim.org/patch.html

3. Installation de J-SIM :

- **Etape4 :**

L'installation du J-Sim se fait sur plusieurs sous étapes:

- On décompresse sous la racine C:\\ le package téléchargé J-Sim_v1.3.tar. On obtient par la suite un dossier nommé J-Sim-1.3. (la décompression des fichiers .tar sous Windows se fait avec le logiciel WINRAR.
- On modifie le fichier setcpath.bat placé sous C:\\jsim-1.3 comme suit :
  - On affecte à la variable **%J_SIM%** le chemin d'installation du simulateur J-SIM
  - On affecte à la variable **% JAVA_HOME %** le chemin d'installation de **J2SDK5.0**
  - On initialise une variable **% ANT_HOME %** au chemin d'installation du compilateur Apache Ant et on ajoute à la variable **%PATH%** le chemin vers les commandes de compilation si on le choisit comme compilateur pour J-SIM.

Si on a choisit de compiler les codes sources du simulateur J-SIM avec la commande « make », on doit donc ajouter le chemin d'installation de à la variable **%PATH%**.

On ajoute à la variable **%CLASSPATH%,** le chemin vers les classes du simulateur J-SIM. Si le compilateur choisi pour le simulateur J-SIM, est Apache Ant, le fichier setcpath.bat doit comprendre les données suivantes :

```
@echo off
set JAVA_HOME=C:\jdk
set J_SIM=C:\jsim-1.3
set ANT_HOME=C:\ant
set PATH=%PATH%;c:\jdk\bin;c:\ant\bin
set CLASSPATH=.;c:\jsim-1.3\classes;c:\jsim-1.3\jars/tcl.zip;c:\jsim-1.3\jars/jython.jar
```

Si la compilation pour le simulateur J-SIM est suivant la commande « make », le fichier setcpath.bat doit être de la forme suivante :

```
@echo off
set JAVA_HOME=C:\jdk
set J_SIM=C:\jsim-1.3
set PATH=%PATH%;c:\jdk\bin;c:\UnxUtils_1\usr\local\wbin
set
CLASSPATH=.;c:\jsim-1.3\classes;c:\jsim-1.3\jars/tcl.zip;c:\jsim- 1.3\jars/jython.jar
```

4. Lancement de J-Sim:

1. On ouvre maintenant la fenêtre des invites des commandes DOS, et on se place sous le répertoire jsim-1.3. Pour cela, on tape les commandes suivantes :

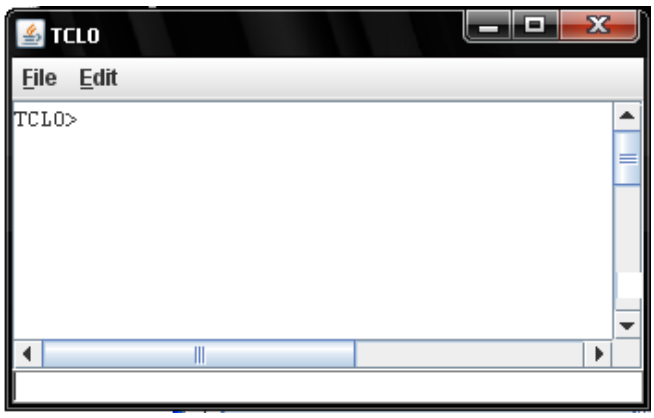
```
cd C:\\

cd jsim-1.3
```

2. On exécute maintenant le fichier setcpath.bat en l'appelant sous la fenêtre d'invite des commandes.
3. On compile tous les codes sources du simulateur J-SIM en tapant la commande « make » si vous avez installé l'utilitaire GNU-Makemakefiles in Java ou la commande « ant compile » si vous avez installé le compilateur Apache Ant.
4. On décompresse maintenant le dossier **patch1.3-4.tar.gz**, déjà téléchargé, sous le répertoire jsim-1.3 et on accepte les mises à jour des fichiers existants.
5. On recompile de nouveau les codes sources et on ignore tous les avertissements.

- **Etape5 :**

Pour vérifier si l'installation de J-SIM a réussi, vous devez avoir une fenêtre pop up intitulé TCL0 qui apparait en tapant la commande suivante : java drcl.ruv.System



- **Etape 6 :**

Chaque fois que vous désirez simuler un script TCL, vous devez changer le chemin courant vers où est placé votre script, puis vous tapez la commande suivante :

javadrcl.ruv.System « nom de votre script.tcl »

5. Conclusion :

Dans cette annexe nous avons présenté la procédure à suivre pour installer et lancer J-Sim sous la plate-forme Windows. Pour le faire sous la plate-forme LINIX vous pouvez consulter le site officiel de J-Sim.