République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique
UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Système Informatique**

Présenté par :

BELKIS Dihia **FERDJI Lydia**

Thème

Proposition d'un protocole de routage hiérarchique Sécurisé pour les réseaux de capteurs sans fil

Mémoire soutenu publiquement le 28/09/ 2016 Devant le jury composé de :

Président : Mr SADOU Samir

Encadreur : Mme BENSAID née BELATTAF Samia

Examinateur : Mr CHEBOUBA Lokmane

Examinateur : Mr DIB Ahmed

REMERCIEMENTS

Grâce à Dieu vers lequel vont toutes les louanges, ce travail s'est accompli. Grâce à Dieu, nous avons l'honneur d'inscrire ici un immense remerciement à nos chers parents.

Nous présentons nos remerciements les plus sincères à l'initiatrice de ce travail Mme Samia BENSAID, qui nous a encadrées avec enthousiasme, pour ses conseils et son soutien qui a fait beaucoup pour la préparation de ce mémoire et nous ont aidées à la finaliser.

Nous présentons nos gratitudes aux membres du jury qui ont bien voulu examiner et évaluer notre travail et qui nous font l'honneur de participer à la soutenance.

Un grand MERCI à tous les enseignants du département informatique qui nous ont formées durant ces cinq dernières années.

Nous aimerions également remercier tous nos amis et camarades de leur soutien et aide et qui nous ont données la force pour continuer.

Nos remerciements vont aussi à tous ceux qui ont contribué de près ou de loin à la concrétisation de ce travail. Qu'ils trouvent tous ici l'expression de notre gratitude et notre parfaite considération.

DEDICACE

Je dédie ce modeste travail

A mes chersparents, Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.

A mes chères sœurs et mon cher frère,

A mes ami(e)s, et toute ma famille,

A ma binôme Dihia et toute sa famille,

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

Lydia

DEDICACE

Je dédie ce modeste travail

A ma très chère maman et mon adorable père,

A mes chères sœurs Cylia et Nina,

A mes chères frères Massi et Jughurta,

A mon fiancé Boudjemaa,

A mes amis, et toute ma famille,

A ma binôme Lydia et toute sa famille,

Et à tous ceux qui m'aiment.

Dihya

Sommaire

Sommaire

Sommaire

Liste des figures

Chapitre I

Chapitre II

Liste des figures

Chapitre IV

RCSF un Réseau de capteurs sans fil.

CH Cluster-Head

CSM Carrier Sense Multiple Access.

CDMA Code Division Multiple Access.

GPS Global Positioning System.

LEACH Low Energy Adaptive Clustering Hierarchy.

MAC Message Authentication Code.

OSI Open System Interconnexion.

SRPBCG Secure Routing Protocol Cluster-Gene-Based for WSNs.

SRPSN Secured Routing Protocol for Sensor Network

SPIN Sensor Protocols for Information via Négociation.

TDMA Time Division Multiple Access.

WSN Wireless Sensor Network.

µTESLA the micro version of the Timed Efficient Stream Loss-Tolerant Authentication Protocol.

NesC Network embedded system C

QoS Quality of Service

TOSSIM TinyOS SIMulator

TinyOS Tiny Open Source

RC4 Rivest Cipher 4

DES Data Encryption Standard

AES Advanced Encryption Standard

RSA Rivest, Shamir et Adleman

SHA Secure Hash Algorithm

MD5 Message Digest 5

ID Identification

SecRoute secure route

IDS Intrusion Detection System

SRPBCG Secure Routing Protocol Cluster-Gene-Based for WSNs.

TL-LEACH Two level Low Energy Adaptive Clustering Hierarchy

E- LEACH Energy Low Energy Adaptive Clustering Hierarchy

LEACH-C Centralized Low Energy Adaptive Clustering Hierarchy

LEACH-F Fixed number of cluster Low Energy Adaptive Clustering Hierarchy

LEACH-A Advanced Low Energy Adaptive Clustering Hierarchy

V-LEACH Vice Cluster Head Low Energy Adaptive Clustering Hierarchy

M- LEACH Mobile Low Energy Adaptive Clustering Hierarchy

LEACH-B Balanced Low Energy Adaptive Clustering Hierarchy

MH- LEACH Multi-Hop Low Energy Adaptive Clustering Hierarchy

I- LEACH Improved Low Energy Adaptive Clustering Hierarchy

Cell- LEACH Cell Low Energy Adaptive Clustering Hierarch

Liste des tableaux et Graphes

Chapitre IV

Chapitre IV

Introduction générale

Durant ces dernières années, nous avons remarqué un développement très rapide des techniques et technologies dans les domaines de l'électronique, la mécanique et les technologies de communication sans fil. Ces innovations ont permis de créer de petits objets communicants appelé capteurs. Ces derniers collaborent entre eux pour former un RCSF capable de superviser une région, et de fournir des informations utiles par la combinaison des mesures prises par les différents capteurs et de les communiquer via des communications multi-sauts, jusqu'à atteindre les stations de base qui sont des points de collecte des données captées. Les stations de base à leur tour, communiquent ces données à l'utilisateur via Internet ou par satellite. Les réseaux de capteurs ont de nombreuses perspectives d'applications dans des domaines très variés : applications militaires, surveillance industrielle ou de phénomènes naturels, santé, …etc.

La problématique de sécurité dans les RCSF est particulièrement importante. En effet, les approches adoptées doivent être différentes des autres solutions proposées pour les autres types de réseaux sans fil. Ceci est dû au fait que les nœuds capteurs sont des composants miniatures ayant des ressources énergétiques et physiques limitées.

Notre projet consiste à sécuriser le protocole de routage MH-LEACH PSM qui est une variante de LEACH, conçu pour les topologies des RCSF hiérarchiques. Notre solution sécurisée doit garantir la sécurité et l'efficacité en performances. Ce mémoire est organisé comme suit :

Le premier chapitre intitulé «La sécurité du routage dans les réseaux de capteurs sans fil» : est une introduction à la sécurité dans les RCSF. Nous y décrirons en général les réseaux de capteurs sans fil. Nous nous intéresserons par la suite à la notion de routage dans ces derniers. Et enfin nous parlerons du routage sécurisé dans ces réseaux.

Dans le deuxième chapitre intitulé « Le protocole de routage hiérarchique LEACH », nous expliquons l'architecture de communication du protocole de routage LEACH et les protocoles MAC utilisés par ce dernier. Nous expliquons en détail l'algorithme et les caractéristiques du protocole LEACH. Nous citons quelques variantes de ce dernier. Par la suite, nous étudions quelques attaques pouvant perturber son fonctionnement et nous donnons quelques solutions existantes.

Dans le troisième chapitre intitulé «Proposition d'un protocole de routage hiérarchique sécurisé », nous proposons notre solution de sécurité pour le protocole MH-LEACH PSM et nous expliquons le schéma de sécurisation et le déroulement de notre nouveau protocole sécurisé.

Le quatrième et dernier chapitre intitulé « Réalisation et simulation », 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 système d'exploitation TinyOS, le langage de programmation NesC et le simulateur TOSSIM.

Pour finir, nous clôturons par une conclusion générale et des perspectives.

1 Introduction

A l'heure actuelle, les réseaux sans fil connaissent une très forte expansion. Ils existent depuis des années, mais l'augmentation de la bande passante et la baisse des coûts ont fait exploser leur croissance.

Ce premier chapitre est consacré à des généralités sur les réseaux de capteurs sans fil (RCSF) et au routage des données dans ces derniers ainsi que leur sécurité. Nous allons aborder dans une première partie des définitions, des notions générales, des caractéristiques, des domaines d'application, l'architecture et les contraintes de conception, ainsi que la pile protocolaire utilisée dans ce type de réseaux. Dans une deuxième partie on va introduire la notion de sécurité des RCSF en présentant les objectifs de la sécurité les différentes attaques qui rendent ce type de réseau vulnérable et les différents mécanismes de sécurité utilisée pour ce type de réseaux .

2 Généralités sur les réseaux de capteur sans fil

2.1 Architecture d'un capteur

Un capteur est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, lumière, pression, humidité, vibration, …, etc.), suivant l'environnement dans lequel il est déployé et l'objectif pour lequel il est conçu, et de la communiquer à un centre de contrôle via une station de base.

Un capteur sans fil est doté, principalement d'une unité de : capture, traitement, communication, stockage et énergie (Comme l'illustre la figureI.1). Des composants additionnels peuvent être ajoutés comme un système de localisation, afin d'identifier la position géographique d'un capteur tel qu'un GPS (Global Position System), un générateur de puissance tel que des cellules solaires afin d'alimenter électriquement le capteur sans avoir à changer ses batteries, ou un mobilisateur pour que les capteurs puissent se déplacer [1] [2].

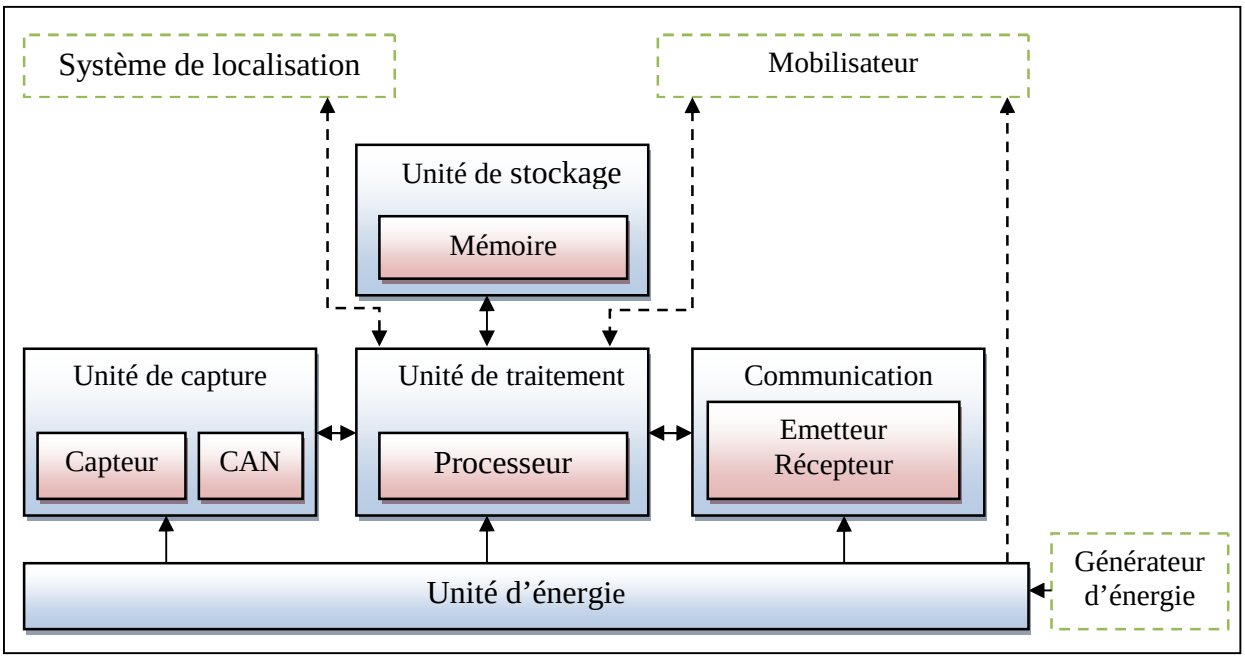


Figure I.1 : Composants d'un capteur.

- **Unité de capture** : chargée de réaliser des mesures physiques (analogiques) sur l'environnement, puis de les convertir en un signal numérique [3].
- **Unité de traitement :** composé principalement d'un processeur et de mémoire (mémoire vive et mémoire non volatile), qui assure le fonctionnement du système d'exploitation, gère les interactions entre les différents modules, et surtout traite les données récoltées [3].
- **Unité de communication :** Elle est responsable des émissions et réceptions des données sur le medium de communication [4].
- **Unité d'alimentation :** Composée généralement d'une ou de plusieurs batteries souvent irremplaçables et non rechargeables ayant des ressources énergétiques limitées [4]. Cette unité est responsable d'alimenter de façon efficace les autres modules du capteur.

2.2 Architecture d'un réseau de capteur sans fil

Le réseau de capteur sans-fil (RCSF) Sensor Network(WSN) en Anglais est composé d'un grand nombre de nœuds dispersé aléatoirement dans une zone géographique dans le but de collecter des informations particulière et de les transmettre par la suite au centre de collecte qui est le nœud puits ou la station de base (En anglais SinkNode) et cela par l'intermédiaire d'une architecture multi-sauts .La station de base transmet ensuite ces données par internet ou par satellite à l'ordinateur central ou centre de traitement des données pour analyse et prise de décision ,comme l'illustre la figure I.2.

Les nœuds ordinaires sont des capteurs, leur type, leur architecture et leur disposition géographique dépendent de l'exigence de l'application en question. Leur énergie est souvent limitée puisqu'ils sont alimentés par des batteries non-rechargeables [34]. Cependant la station de base est un nœud particulier du réseau. Il est chargé de la collecte des données issues des différents nœuds du réseau. Il doit être toujours actif puisque l'arrivée des informations est aléatoire. C'est pourquoi son énergie doit être illimitée. Dans un réseau de capteur sans fils plus ou moins large et à charge un peu élevée, on peut trouver deux nœuds puits ou plus pour alléger la charge. [34]. Il y a essentiellement trois types de station de base [33]:

- Un nœud appartenant au réseau comme n'importe quel autre nœud.
- Une entité extérieure au réseau. Pour ce deuxième cas, la station de base peut être un dispositif extérieur, par exemple, un ordinateur portable interagissant avec le réseau.
- Une passerelle vers un autre réseau tel qu'Internet, où la demande de l'information vient d'un certain centre de traitement lointain.

Le centre de traitement des données est le terminal vers lequel les données collectées par la station de base sont envoyées. Son rôle est de regrouper les données issues des nœuds et les traiter de façon à en extraire de l'information utile exploitable. Le centre de traitement peut être éloigné de la station de base, alors les données doivent être transférées à travers un autre réseau, c'est pourquoi on introduit une passerelle entre la station de base et le réseau de transfert pour adapter le type de données au type du canal [33].

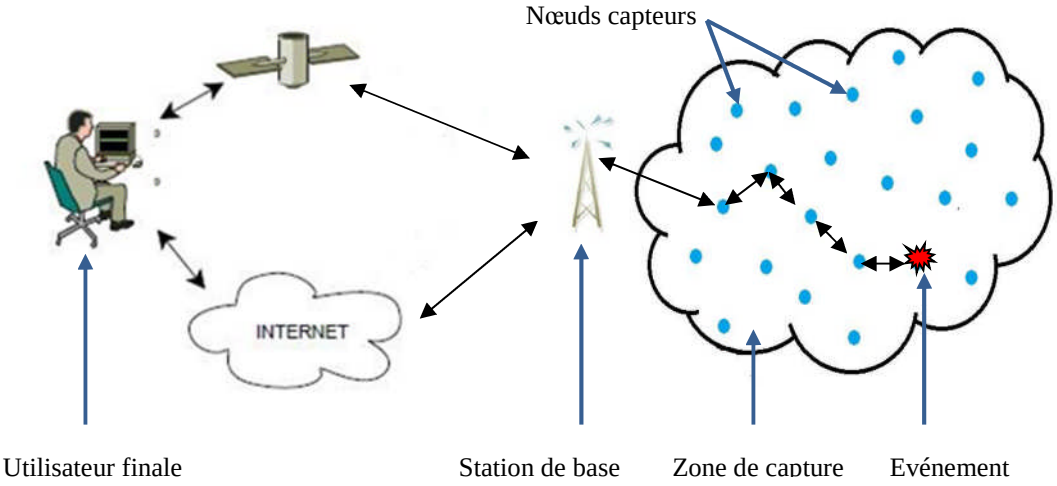


Figure I.2 : Réseau de capteur sans fil.

2.3 Types de réseaux de capteurs sans fil

Il existe deux grands types de réseaux de capteurs sans fil :

- Soit le réseau est constitué d'un ensemble de capteurs mobiles. Le but de tels réseaux est la plupart du temps l'exploration de zones inaccessibles ou dangereuses. Les travaux de recherche sont souvent orientés robotique, les nœuds jouant à la fois le rôle de capteur et d'actionneur.
- Soit le réseau est constitué de capteurs fixes servant à la surveillance d'occurrence d'évènements sur une zone géographique Ici, le réseau n'effectue que la surveillance, les données mesurées sont transmises en mode multi-sauts à la station de base.

2.4 Caractéristiques d'un réseau de capteur sans fil

Parmi les caractéristiques les plus importantes d'un réseau de capteurs, nous citons [8] **:**

- **La durée de vie limitée :**
  Les nœuds capteurs sont très limités par la contrainte d'énergie, ils fonctionnent habituellement sans surveillance dans des régions géographiques éloignées. Par conséquent, recharger ou remplacer leurs batteries devient quasiment impossible.
- **Ressources limitées :**
  Habituellement, les nœuds capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être mises dans ces nœuds. En conséquence, la capacité de traitement et de mémoire est très limitée.
- **Topologie dynamique :**
  La topologie des réseaux de capteurs change d'une manière fréquente et rapide car les nœuds capteurs peuvent être déployés dans des environnements hostiles (par exemple un champ de bataille), la défaillance d'un nœud capteur peut donc être très probable. De plus, les nœuds capteurs et les nœuds finaux où ils doivent envoyer l'information capturée peuvent être mobiles.
- **Agrégation des données :**
  Dans les réseaux de capteurs, les données produites par les nœuds capteurs sont très reliées, ce qui implique l'existence de redondances de données. Une

approche répandue consiste à agréger les données au niveau des nœuds intermédiaires afin de réduire la consommation d'énergie lors de la transmission de ces données

- **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.
- **Bande passante limitée :**
  Une des caractéristiques primordiales des réseaux de capteurs basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un nœud est limitée.
- **Sécurité physique limitée :**
  Les réseaux de capteurs sans fil 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.

2.5 Domaines d'application des réseaux de capteurs sans fil

Les réseaux de capteurs sans fil RCSF ont un champ d'application vaste et diversifié. Ceci est rendu possible par leur cout faible, leur taille réduite, le support de communication sans fil utilisé et la large gamme des types de capteurs disponibles. Un autre avantage est la possibilité de s'auto-organiser et d'établir des communications entre eux sans aucune intervention humaine, notamment dans des zones inaccessibles ou hostiles, ce qui accroit davantage le nombre de domaines ciblés par leur application (environnement, catastrophes naturelles, bâtiments intelligents, la santé, l'agriculture, l'industrie…etc.). Nous présentons dans ce qui suit les domaines les plus ciblés par les RCSF [6] :

- **Domaine militaire**: les RCFS sont utilisés pour la surveillance, la détection des intrusions, la détection des substances dangereuses, la communication, la reconnaissance et le ciblage.
- **Domaine commercial**: Il est possible d'intégrer des nœuds capteurs au processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la position actuelle du paquet.
- **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.
- **Domaine médical**: parmi ses applications, on peut citer la surveillance, l'état des patients et le taux de médicaments qui leur ont été administrés, et l'aide à la localisation des médecins et des patients au sein d'un hôpital.
- **Domaine architectural**: transformation des bâtiments en environnements intelligents capables de reconnaitre des personnes, interpréter leurs actions et y réagir.

2.6 Pile protocolaire

Les réseaux de capteurs sans fil imposent des contraintes supplémentaires aux protocoles de communication. Par conséquent, le modèle traditionnel en couches (modèle OSI), ne répond pas aux exigences de ce type particulier de réseaux [8]. En effet, dans le but d'un établissement efficace d'un RCSF, une architecture en couches est adoptée afin d'améliorer la robustesse du réseau. Une pile protocolaire de cinq couches similaires à celles du modèle OSI (physique, liaison, réseau, transport et application) [2][7] est donc utilisée par les nœuds du réseau (Figure I.3).

- **Couche application :** Elle assure l'interface avec l'utilisateur, son rôle est d'implémenter l'ensemble d'applications et de logiciels d'interaction.
- **Couche transport :** Son rôle est le contrôle du flux, le découpage, l'ordonnancement et le transport des paquets de données, et la gestion des erreurs de transmission.
- **Couche réseau** : L'objectif de cette couche est de trouver des chemins de routage à faible coût d'énergie pour transmettre les données captées vers la station de base. Ainsi, les protocoles de cette couche doivent toujours prendre en compte les limitations en ressources des nœuds capteurs.
- **Couche liaison de données** : Cette couche est chargée du contrôle d'erreurs, du multiplexage des flux de données, et le contrôle d'accès au média de transmission. Comme l'environnement des réseaux de capteurs est bruyant et les nœuds peuvent être mobiles, la couche de liaison de données doit garantir une faible consommation d'énergie et minimiser les collisions entre les données diffusées par les nœuds voisins.
- **Couche physique :** Comme celle du modèle OSI, cette couche est responsable de la modulation, la détection du signal et la sélection des fréquences porteuses. Elle doit assurer des techniques d'émission, de réception et de modulation de données d'une manière robuste.

De plus, cette pile possède trois plans (niveaux) de gestion dédiés pour le contrôle d'énergie, de mobilité et des tâches particulières :

- **Plan de gestion d'énergie :** Il contrôle l'utilisation de la batterie. Comme la vie du nœud a une dépendance forte à l'égard de la vie de la batterie, il doit par conséquent contrôler et minimiser sa consommation d'énergie. Par exemple, après la réception d'un message, le capteur éteint son récepteur et se met en mode sommeil afin d'éviter la duplication des messages déjà reçus. En outre, si le niveau d'énergie atteint un seuil bas, le nœud diffuse à ses voisins une alerte pour les informer qu'il ne peut pas participer au routage, l'énergie restante est réservée à la capture
- **Plan de gestion de mobilité :** Il détecte et enregistre le mouvement des nœuds capteurs afin de maintenir des informations sur leurs localisations et d'entretenir continuellement une route vers l'utilisateur final.
- **Plan de gestion de tâches :** Il équilibre et ordonnance les différentes tâches de captage de données dans une région spécifique. Il n'est pas nécessaire que tous les nœuds de cette région effectuent la tâche de captage au même temps, certains nœuds exécutent cette tâche plus que d'autres selon leur niveau de batterie [2].

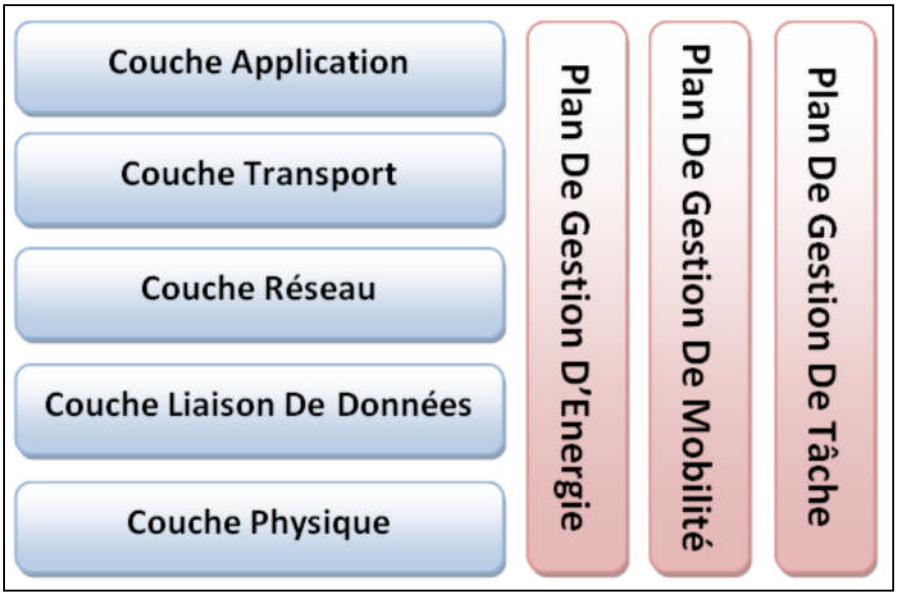


Figure I.3 : La pile protocolaire dans les RCSF.

3 Routage dans les réseaux de capteurs

3.1 Classification des réseaux de capteurs

Récemment, les protocoles de routage pour les RCSF ont été largement étudiés. Ils peuvent être classifiés selon plusieurs critères, comme le montre la figure I.4:

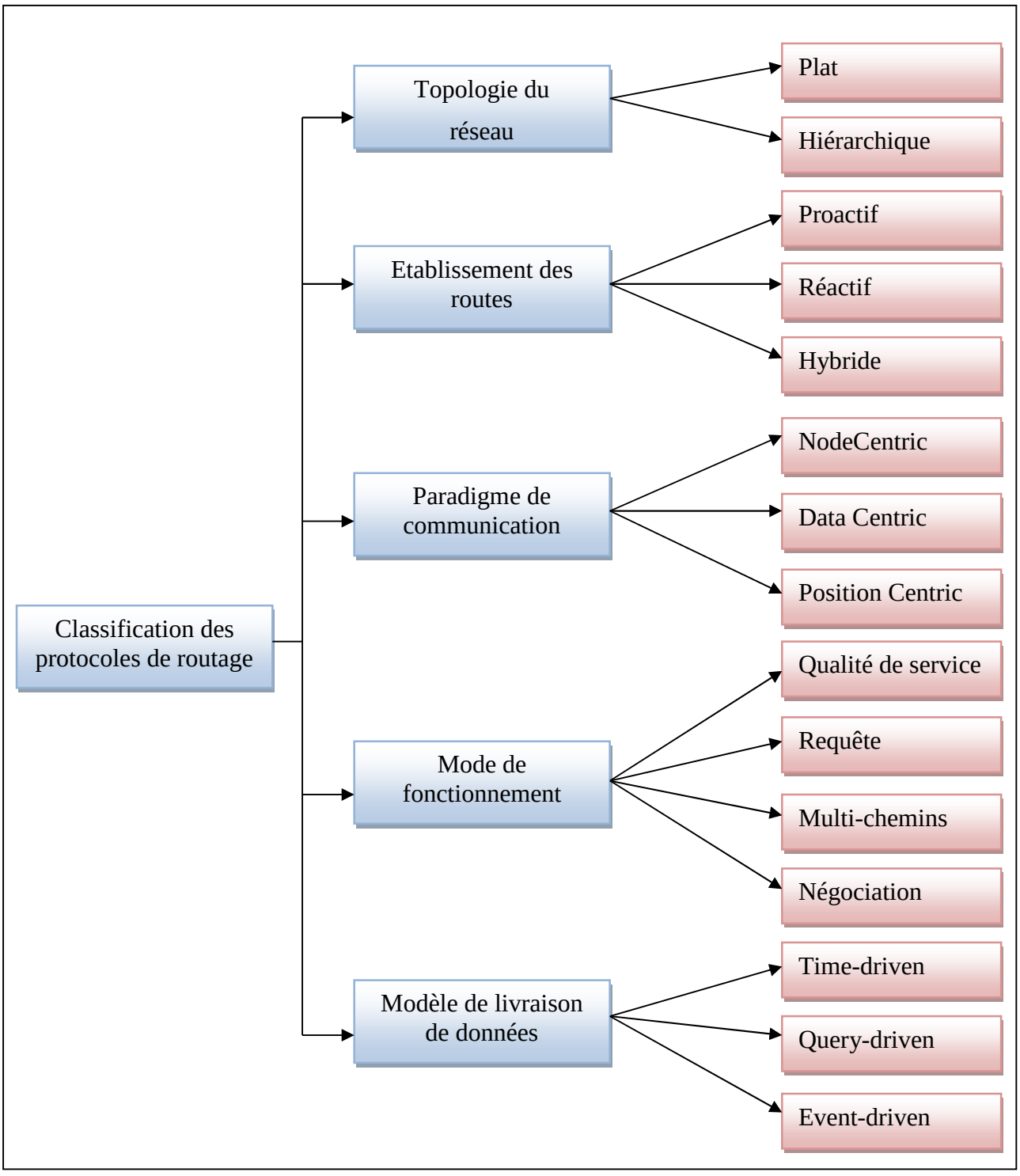


Figure I.4 : Classification des protocoles de routage dans les RCSF [52].

3.1.1 Selon la topologie du réseau

La topologie détermine l'organisation des capteurs dans le réseau. Globalement, il existe deux topologies dans les RCSF [24]: La topologie plate et la topologie hiérarchique. Comme on le distingue sur la figure I.4

- **Topologie plate :**
  Les protocoles à topologie plate (flat) considèrent que tous les nœuds sont semblables en termes de ressources et ils possèdent tous le même rôle excepté le nœud « station de base » qui est chargé de la collecte des données issues des différents nœuds capteurs afin de les transmettre vers les centres de traitement. [29]
- **Topologie hiérarchique**
  Le routage hiérarchique [27][28] est considéré comme étant l'approche la plus favorable en termes d'efficacité énergétique. La topologie hiérarchique divise les nœuds en plusieurs niveaux de responsabilité. L'une des méthodes les plus employées est le clustering, où le réseau est partitionné en groupes appelés "clusters". Un cluster est constitué d'un chef (cluster-head) et de ses membres.

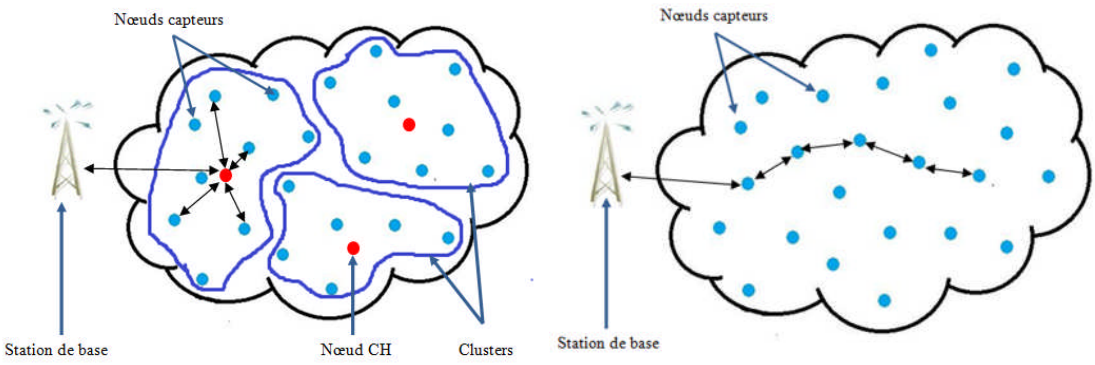


Figure I.5 : Topologie plate à droite et clustérisé à gauche d'un RCSF [30].

Un réseau basé sur une topologie hiérarchique doit avoir au moins trois niveaux dans sa hiérarchie, puisqu'un réseau avec un nœud central Sink et seulement un niveau hiérarchique au-dessous, forme une topologie en étoile. À titre d'exemple des protocoles utilisant une topologie hiérarchique on peut citer le Protocol **LEACH** (Low energy Adaptive Clustering Hierarchy) .

3.1.2 Selon la méthode d'établissement de routes

Suivant la manière de création et de maintien des chemins pendant le routage, nous distinguons trois catégories de protocoles de routages : protocoles proactifs, réactifs ou hybrides [25].

- **Protocole proactif**
  Dans cette catégorie dite à diffusion de tables, les protocoles de routage maintiennent à jour une table de routage dans chaque nœud. A chaque changement de la topologie du réseau, des messages de mise à jour sont communiqués aux nœuds. Les protocoles proactifs sont adaptés aux applications qui nécessitent un prélèvement périodique des données. Et par conséquent, les capteurs peuvent se mettre en veille pendant les périodes d'inactivité, et n'enclencher leur dispositif de capture qu'à des instants particuliers.
- **Protocole réactif**
  Les protocoles de routage appartenant à cette catégorie se basent sur la découverte et le maintien des routes. Suite à un besoin, un nœud lance une procédure de découverte de route en diffusant un paquet de contrôle (Route Request) dans le réseau à la

recherche d'une route vers le destinataire, ce processus (Route Request) s'arrête une fois la route trouvée ou toute les possibilités sont examinées. Dès que la communication est établie, cette route est maintenue jusqu'à ce que la destination devienne inaccessible ou jusqu'à ce que la route ne soit plus désirée. Ce type de protocoles est pratique pour des applications temps réel où les capteurs doivent réagir immédiatement à des changements soudains des valeurs captées.

- **Protocole hybride**
  Les protocoles hybrides combinent les deux idées: celle des protocoles proactifs et celle des protocoles réactifs. Ils utilisent un protocole proactif pour avoir des informations sur les voisins les plus proches (au maximum les voisins à deux sauts). Au-delà de cette zone prédéfinie, le protocole hybride fait appel aux techniques des protocoles réactifs pour chercher des routes.

3.1.3 Selon les paradigmes de communication

Le paradigme de communication détermine la manière dont les nœuds sont interrogés. Dans le RCSFs, il existe trois paradigmes de communication [26].

- **Nodecentric**
  Ce paradigme est celui employé dans les réseaux conventionnels, où les communications se basent sur l'identification des nœuds participants à l'aide d'adresses IP.
- **Data centric**
  Dans ce paradigme, les communicants sont identifiés par leurs données. Ainsi, le système peut être vu comme une base de données distribuée, où les nœuds forment des tables virtuelles, alimentées par les données captées.
- **Position centric**
  Dans ce paradigme, le système est interrogé en utilisant la position des nœuds. Dans ce cas, le routage s'effectue grâce à des techniques géométriques afin d'acheminer l'information d'une zone géographique vers une autre.

3.1.4 Selon le mode de fonctionnement du protocole

Le mode de fonctionnement définit la manière avec laquelle les données sont propagées dans le réseau. Selon ce critère, les protocoles de routage peuvent êtres classifiées en quatre catégories: routage basé sur la qualité de service "QoS" routage base sur les requêtes, routage multi-chemins, et routage basé sur la négociation [24].

- **Routage basé sur la qualité de service**

Dans les protocoles de routage basé sur la QoS, le réseau doit équilibrer entre la consommation d'énergie et la qualité de données. En particulier, le réseau doit satisfaire certaines métriques de QoS, par exemple, retard, énergie, largeur de bande passante, etc. Les protocoles de cette approche sont très recommandés pour les applications de surveillance (centrales nucléaires, applications militaires, etc).

- **Routage basé sur les requêtes**

Dans ce type de routage, le puits génère des requêtes afin d'interroger les capteurs. Ces requêtes sont exprimées soit par un schéma valeur-attribut ou bien en utilisant un langage spécifique (par exemple SQL : Structured Query Language). Les nœuds qui détiennent les données requises doivent les envoyer au nœud demandeur à travers le chemin inverse de la

requête. Les requêtes émises par le puits peuvent aussi être ciblées sur des régions spécifiques de réseau.

- **Routage basé sur les multi-chemins**

Afin d'augmenter la performance du réseau, le routage à chemins multiples consiste à considérer non plus une seule route entre une source et une destination pour la transmission de données, mais un ensemble de chemins. Cette technique répond à trois objectifs principaux : la tolérance aux pannes, le partage de charge, et l'augmentation de la bande passante [31].

- **Routage basé sur la négociation**

En détectant le même phénomène, les nœuds capteurs inondent le réseau par les mêmes paquets de données. Ce problème de redondance peut être résolu en employant des protocoles de routage basés sur la négociation. En effet, avant de transmettre, les nœuds capteurs négocient entre eux leurs données en échangeant des paquets de signalisation spéciales, appelés META-DATA. Ces paquets permettent de vérifier si les nœuds voisins disposent des mêmes données à transmettre [32]. Cette procédure garantit que seules les informations utiles seront transmises et élimine la redondance des données.

3.1.5 Selon le modèle de livraison de données

Il est possible de distinguer trois modèles de livraison de données : time-driven, query-driven et event-driven [22][23] :

- **Time-driven**

Un réseau time-driven est approprié pour des applications qui nécessitent un prélèvement périodique des données. Par exemple, cela est utile dans des applications de monitoring (feu, météo) afin d'établir des rapports périodiques.

- **Query-driven**

Dans les applications query-driven, la collecte d'informations sur l'état de l'environnement et la livraison des données sont initiées par des requêtes envoyées généralement par le nœud puits.

- **Event-driven**

Dans des applications temps réel, les capteurs doivent réagir immédiatement à des changements soudains des valeurs captées. Dans ce cas, le protocole de routage doit être réactif et doit donner des réponses rapides à l'occurrence d'un certain nombre d'évènements.

3.2 Facteurs de conception des protocoles de routage

La conception des RCSF est influencée par plusieurs contraintes. Ces facteurs importants servent comme directives pour le développement des algorithmes et protocoles utilisés dans les réseaux de capteurs, ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine [16].

- **Tolérance aux pannes**

Le but de la tolérance aux pannes est d'éviter la faille totale du système malgré la présence de fautes dans un sous ensemble de ses composants élémentaires [10]. Le réseau doit être capable de maintenir ses fonctionnalités sans interruptions en cas de défaillance d'un ou plusieurs de ses capteurs (à cause d'un épuisement d'énergie, ou subissent des dommages

physiques). De ce fait, des protocoles et des techniques peuvent être conçus pour évaluer le niveau de la tolérance aux pannes exigé dans les réseaux de capteurs (la tolérance aux pannes exigée peut être basse si les capteurs sont déployés dans un habitat alors que si les capteurs sont déployés dans un champ de bataille, la tolérance aux pannes devrait être élevée) [11].

- **Consommation d'énergie**

Le facteur le plus important à prendre en considérations est l'énergie consommée par un capteur lors de la détection et de la transmission des données captées sur le réseau. Du fait que la transmission est la fonction qui consomme le plus d'énergie [12] et que le remplacement des batteries est impossible dans la plupart des cas. Pour augmenter la durée de vie d'un réseau, les chercheurs ont opté pour des techniques qui favorisent le traitement local des données afin de réduire la taille du paquet. Ces techniques évitent la redondance des informations à transmettre et qui mettent le capteur en mode sommeil le plus longtemps possible [13][14].

- **Limitation des capacités des nœuds**

Un capteur est très limité, en ce qui concerne, les traitements locaux à cause de sa taille minimale (fonctionne dans la majorité des cas avec des registres 8 ou 16 bits, une RAM de 2 à 250 Ko et une mémoire flash de 1 à 32 Mo) [17]. Cela signifie que le protocole de routage doit être simple et peu exigeant en capacité de calcul et de stockage.

- **Passage à l'échelle (Scalability)**

Les RCSF sont généralement déployés avec un grand nombre de capteurs qui peut atteindre le million. Ceci peut engendrer des problèmes de communication et de contrôle [8], donc les protocoles de routage doivent être très scalables . Autrement dit, les protocoles de routage ne devraient pas souffrir d'une dégradation de performances dans le cas d'endommagement de nœuds aussi bien qu'avec un nombre plus élevé de nœuds [15].

- **Les couts de production**

Souvent les réseaux de capteurs sont composés d'un très grand nombre de nœuds. Le prix d'un nœud est critique afin de pouvoir concurrencer un réseau de surveillance traditionnel.

- **Environnement**

Les capteurs sont souvent déployés en masse dans des endroits tels que des champs de bataille au-delà des lignes ennemies, à l'intérieur de grandes machines, au fond d'un océan, dans des champs biologiquement ou chimiquement hostiles [18]. Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiques éloignées [11].

- **Topologie du réseau**

Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. La plupart des architectures réseau reposent sur des capteurs statiques. Pourtant, la mobilité des stations de base et/ou des capteurs est parfois nécessaire dans de nombreuses applications. Egalement, l'état d'activité des capteurs (extinction, mise en veille et actif) intervient dans les changements de la topologie. La conception d'un protocole d'auto-organisation qui s'adapte continuellement et rapidement aux changements, s'avère nécessaire pour assurer le bon fonctionnement du réseau [19].

- **Qualité de service**

Dans un réseau de capteurs, les nœuds sont reliés par une architecture sans fil. Pour permettre des opérations sur ces réseaux dans le monde entier, le média de transmission doit être

standardisé. On utilise le plus souvent l'infrarouge, le Bluetooth [20] et les communications radio Zig Bee[53], WIFI [21].

4 Sécurité dans les réseaux de capteurs sans fil

Dans ce qui suit on va détailler les principaux points qui concernent la sécurité des RCSF

4.1 Objectifs de la sécurité dans les réseaux de capteurs sans fil

La sécurité des informations transitant dans les RCSF doivent répondre à plusieurs prés requis.

- **Disponibilité du réseau :**
  Le réseau doit pouvoir être disponible à tout instant, c'est à dire que l'envoi d'information ne doit pas être interrompu, de même que la circulation de l'information ne doit pas être stoppée [14] .Cette propriété reste difficile à assurer dans les RCSF étant donné les contraintes qui pèsent sur ces réseaux, à savoir : topologie dynamique, ressources limitées des nœuds de transit ainsi que des communications sans fil pouvant être facilement brouillées ou perturbées [13].

- **Authentification :**
  L'authentification des capteurs est nécessaire pour s'assurer que l'identité déclarée par un capteur est bien celle du capteur déclarant. En l'absence d'un mécanisme permettant d'authentifier clairement un nœud du réseau, de nombreuses attaques peuvent se mettre en place comme l'attaque Sybil [14].

- **Intégrité des données :**
  Les données circulant sur le réseau ne doivent pas pouvoir être altérées au cours de la communication. Il faut donc s'assurer que personne ne puisse capturer et modifier les données du réseau. De la même manière il faut vérifier que les données n'ont pas subi d'altération due à un disfonctionnement du matériel, qui est un risque important sur des capteurs sensibles aux altérations d'états [14]. L'intégrité peut être assurée par l'utilisation des fonctions de hachage cryptographiques qui permettent d'obtenir pour chaque message une empreinte numérique [13].

- **La confidentialité :**
  La confidentialité reste un point important, étant donné la communication sans fil des RCSF. Elle consiste à préserver le secret des messages échangés et ne pas les révélés aux adversaires. La confidentialité peut être assurée par l'usage de la cryptographie à clé symétrique ou asymétrique [13].

- **Fraîcheur des données :**
  La fraîcheur des données permet de savoir si la donnée est récente ou non. Cela signifie qu'il faut s'assurer que la donnée transmise correspond à un état présent. La fraîcheur des données garantit ainsi que ces données ne reflètent pas un état passé qui n'a plus cours [15].

- **Auto organisation :**
  Les capteurs du réseau doivent être capables, après avoir été déployés, de s'auto organiser et surtout de se sécuriser eux-mêmes, sans autres interventions extérieures [15]

- **Localisation sécurisée :**
  Le besoin de se localiser et de connaître la position des autres nœuds peut être primordial dans de nombreux cas pour déjouer d'éventuelles attaques jouant sur les distances [14].

4.2 Vulnérabilités des réseaux de capteurs sans fil

Les vulnérabilités liées à la sécurité dans les réseaux de capteurs sans fil se résument en :

- Nœuds compromis **:** Les nœuds capteurs sont plus faciles à compromettre que ceux dans un réseau traditionnel, parce qu'ils sont de nature mobile et sans-fil donc physiquement plus petits et plus faciles à déplacer et à attaquer. [16]

- La technologie sans fil sous-jacente. Quiconque possédant le récepteur adéquat peut potentiellement écouter ou perturber les messages échangés [5]

- Les nœuds eux-mêmes sont des points de vulnérabilité du réseau car une attaque peut compromettre un composant laissé sans surveillance [5]

- Faible capacité, ou nœuds hétérogènes : La capacité souvent limitée des nœuds et l'utilisation de batteries pour l'alimentation des équipements sont aussi des faiblesses des réseaux de capteurs. Ainsi, les nœuds capteurs ont une durée de vie limitée [16]

- L'absence d'infrastructure fixe pénalise l'ensemble du réseau dans la mesure où il faut faire abstraction de toute entité centrale de gestion pour l'accès aux ressources [5]

- Les mécanismes de routage sont d'autant plus critiques dans les RCSFs que chaque nœud participe à l'acheminement des paquets à travers le réseau. De plus, les messages de routage transitent sur les ondes radio cela rend les attaques sur le réseau de capteurs tels que l'accès au réseau, interception et déni de service plus faciles à achever que dans les réseaux filaires.

4.3 Attaques dans les réseaux de capteur sans fil

Une attaque peut être définie comme une tentative d'accès non autorisé à un service, une ressource ou une information, ou bien la tentative de compromettre l'intégrité, la disponibilité, ou la confidentialité du réseau [40].

Les capteurs peuvent être objets de plusieurs types d'attaques, suivant leurs fonctionnements et leurs localités dans le réseau. On cite ci-dessous les principaux objectifs des attaques sur un RCSF :

- **Espionnage** : l'attaquant cherche à obtenir les données du réseau dont l'accès n'est pas autorisé. L'espionnage peut s'effectuer par une écoute passive ou en envoyant des requêtes aux nœuds capteurs, aux agrégateurs ou par la capture des nœuds pour avoir plus d'informations
- **Perturbation** : le but de l'attaquant est de perturber le fonctionnement du réseau par injection des données erronées, altération des messages de données ou par manipulation directe de l'environnement en générant des fausses alertes .
- **Détournement** : dans ce cas, l'attaquant cherche à détourner les applications des capteurs de leurs bons fonctionnements, par l'obtention du contrôle d'un ou d'un

ensemble de capteurs. L'obtention du contrôle peut se faire, par exemple, par la compromission ou la capture d'un nœud.

Dans la littérature, ces différentes attaques ont été classifiées de diverses manières [45][44].

4.3.1 Classification des attaques selon l'origine

- **Attaques internes**

Elle se produit à l'intérieur du réseau. Dans ce cas, l'intrus est aperçu par les autres nœuds comme étant un nœud normal. Ce phénomène ce produit lorsque le nœud malveillant connait la clé de chiffrement et peut enclencher le processus de cryptage et décryptage. Par conséquent, il peut accéder aux messages chiffrés échangés entre les nœuds. L'attaque interne est considérée comme la plus dangereuse du point de vue sécurité.

- **Attaques externes**

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

4.3.2 Classification des attaques selon la nature

- **Attaques passives**

Dans ce type d'attaques, l'attaquant passe inaperçu. Ce type d'attaque est généralement indétectable mais une prévention est possible. L'objectif de ces attaques est d'intercepter les données circulant sur le réseau afin d'extraire des informations importantes, et d'analyser les chemins empruntés par ces données afin d'obtenir des informations sur le fonctionnement global du réseau (les positions des stations de bases…etc.). Ensuite, Les informations obtenues par un attaquant passif peuvent être utilisées pour créer des attaques actives [42] [43].

- **Attaques actives**

Contrairement aux attaques passives, les attaques actives visent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables.

Les menaces actives appartiennent principalement à quatre catégories (illustrées par la figure I.6) :

- Interruption : Vise la disponibilité des données (ex. : coupure d'une ligne de communication).
- Interception : Vise la confidentialité des données (ex : copie non autorisée de fichiers ou de programmes).
- Modification : Vise l'intégrité des données (ex : modifier le contenu de messages transmis sur un réseau).
- Fabrication : Vise l'authenticité des données (ex : insertion de faux messages dans un réseau).

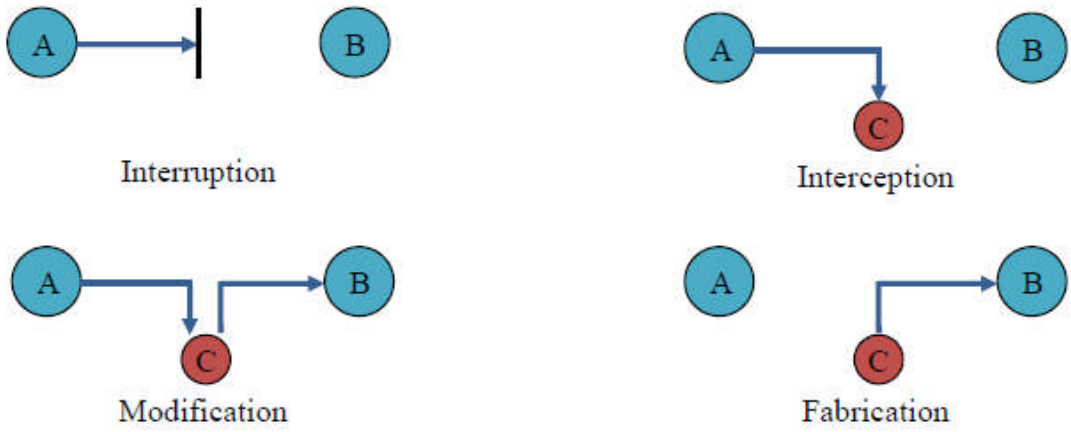


Figure I.6 : Types d'attaques actives

4.4 Mécanismes de sécurité dans réseaux de capteurs sans fil

Un mécanisme est conçu Pour contrer les attaques qui menacent les réseaux de capteurs sans fil. Le développement d'un mécanisme de sécurité est souvent confronté à la nature limitée des ressources dans les RCSF. Ainsi, on doit toujours faire un compromis entre la sécurité assurée et le surcoût introduit par la contre-mesure appliquée. Basés sur la puissance et le degré de malveillance de l'attaquant, plusieurs dispositifs de contre-mesure ont été proposés. Cependant, la plupart de ces mécanismes supposent rarement des modèles d'attaquants à grande puissance, d'où la nécessité de les combiner afin de satisfaire toutes les conditions de sécurité. La section suivante présente les différents types des contre-mesures disponibles.

4.4.1 Partitionnement de données :

C'est une solution proposée dans le but d'empêcher les attaquants de récupérer l'intégralité des informations qui circulent sur le réseau. Le principe est de découper, au niveau du nœud source, l'information à transmettre en plusieurs paquets de tailles fixes et chaque paquet de données sera envoyé sur un chemin différent. L'entité de destination finale, et après la réception de tous les paquets de données, procède à la reconstitution du message initial émis par le nœud source. Néanmoins, cette solution est gourmande en énergie car elle implique un grand nombre de nœuds pour acheminer chaque paquet vers l'entité destinatrice. La figure I.7 schématise un partitionnement de données dans un RCSF.

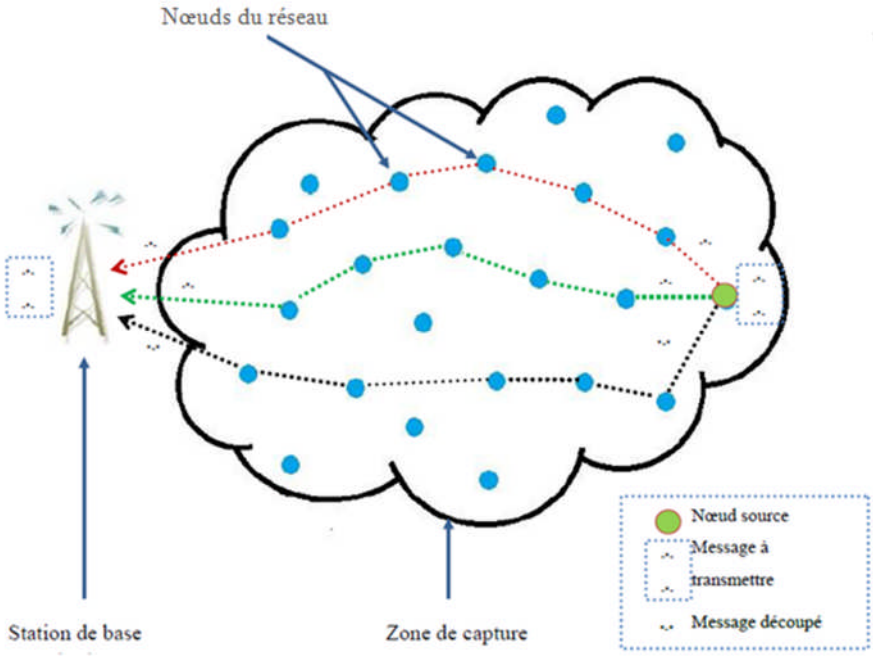


Figure I.7. Technique de partitionnement de données

4.4.2 La Cryptographie

Le mot « cryptographie » est composé des mots grecques: « crypto » signifie caché, «graphy » signifie écrire. C'est donc l'art de l'écriture secrète [46]. Elle est définie comme étant une science permettant de convertir des informations "en clair" en informations cryptées (codées), c'est à dire non compréhensibles, et puis, à partir de ces informations cryptées, de restituer les informations originales.

a) Le chiffrement :

Le chiffrement est un système basé sur des clés cryptographique assurant la confidentialité. On distingue deux classes de primitives : symétrique ou asymétrique [4].

- **Le chiffrement symétrique :**

  La Figure I.8 schématise le principe de chiffrement symétrique qui se base sur le partage d'une même clé pour chiffrer et déchiffrer les messages échangés par une paire de nœuds. Le chiffrement symétrique se fait soit en fractionnant les données en bit à bit (chiffrement par flots) ou bien en blocs de taille fixe (chiffrement par blocs).

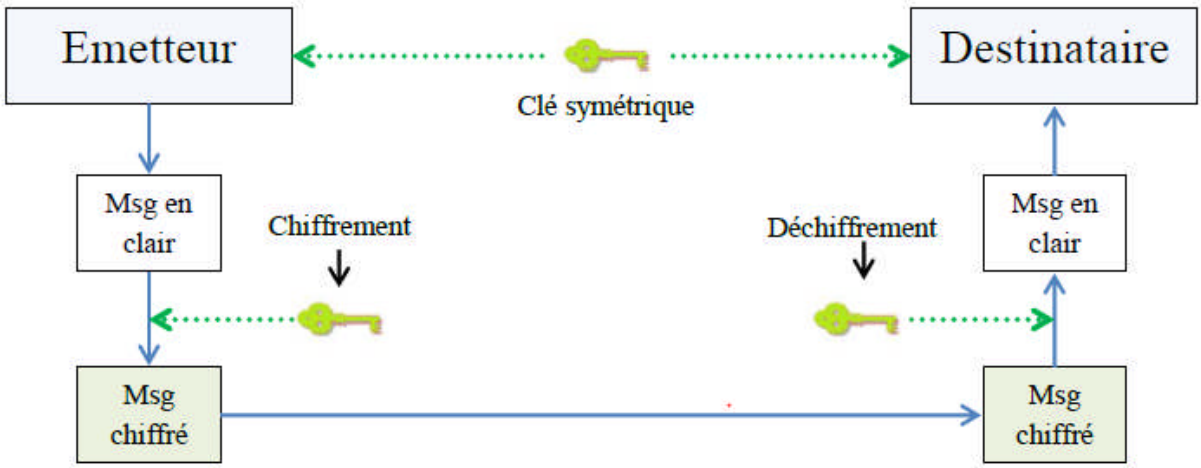


Figure I.8. : Chiffrement symétrique

- **Le chiffrement asymétrique**

Dans ce cas, deux clés sont utilisées, une clé publique et une clé privée. Lorsqu'un émetteur envoie un message, il le chiffre avec la clé public du destinataire, le nœud récepteur génère deux clés différentes une clé privé pour déchiffrer le message reçu et une clé publique diffusé à ces voisins, comme le montre bien la figure I.9. L'algorithme de chiffrement asymétrique le plus connu est : RSA (Rivest Shamir Adleman) [50].

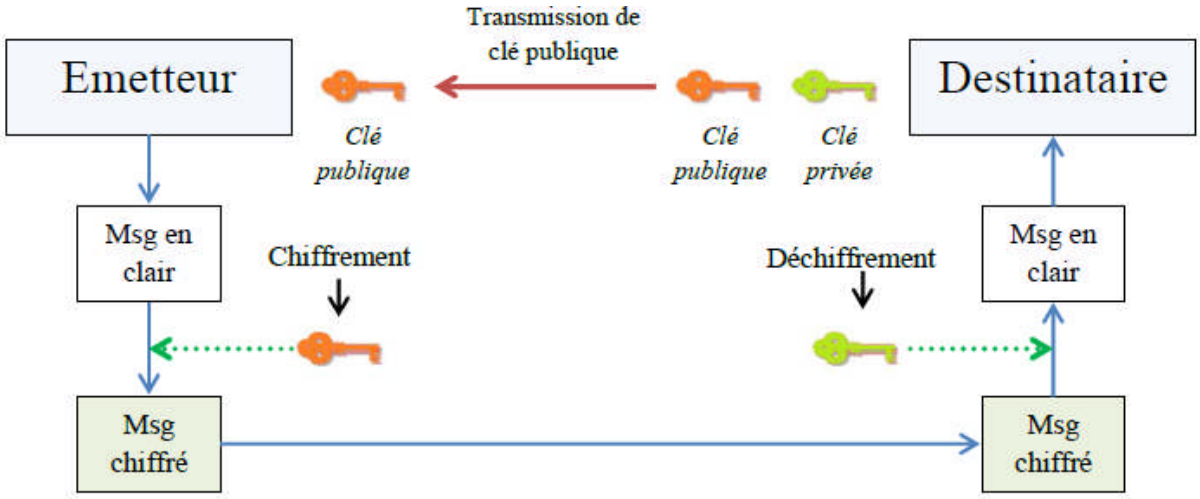


Figure I.9. : Chiffrement asymétrique

- **Chiffrement par flot (en chaine):**

Le chiffrement en chaîne est fait bit à bit sans attendre la réception entière des données. L'algorithme le plus connu est : RC4 (Rivest Cipher 4) [47].

- **Chiffrement par blocs :**

Le chiffrement par bloc consiste à fractionner les données en blocs de taille fixe (64 bits, 128 bits). Chaque bloc sera ensuite chiffré une fois qu'il atteint la taille envisagée. C'est la méthode la plus utilisé dans les systèmes cryptographiques. Les

algorithmes les plus connus sont : DES (Data Encryption Standard) [48], AES (Advanced Encryption Standard) [49].

b) La gestion de clés

C'est une technique de sécurité simple utilisant le principe de clé, occupant une taille mémoire minimal et facile à calculer. Elle peut définir quatre types de clés [4] :

- **Clé globale :** une clé unique est partagée par tous les nœuds du réseau. Tous les messages peuvent être chiffrés avec la même clé. Cette solution est très économe en énergie mais moins sure et moins sécurisante.
- **Clé partagé par paire de nœud :** chaque nœud possède une clé différente pour communiquer avec un nœud voisin qui partage cette clé. Ainsi si un nœud possède "n" voisins, il aura "n" clés à stocker pour pouvoir communiquer avec ses voisins. Dans cette solution, un nœud qui cherche à envoyer un message, doit l'encrypter avec la clé du voisin qui recevra l'information. Le nœud voisin devra déchiffrer l'information pour la chiffrer à nouveau avec la clé qui correspond au destinataire suivant. C'est la solution cryptographique la plus sécurisée mais aussi la plus coûteuse en terme d'énergie et de latence.
- **Clé partagé par groupe de nœuds :** cette solution combine les deux premières techniques, et elle apporte un compromis entre sécurité et consommation d'énergie. Dans ce cas, les communications entre les nœuds d'un même cluster se font avec une clé globale, et les chefs de groupes communiquent entre eux soit avec une clé commune à tous les chefs de groupes, soit une clé partagée par paire de chefs de groupe.
- **Clé individuelle :** chaque nœud possède une clé personnelle pour chiffrer son information. Cette clé est partagée uniquement avec la station de base. Lorsque l'émetteur chiffre son message avec sa clé privé, le message circulera d'une manière caché sur le réseau jusqu'à atteindre la station de base. Mais cette solution reste inappropriée pour les architectures où on effectue des traitements sur les données (exemple : agrégation de données).
  La cryptographie est réalisée selon certains outils utilisant le principe de clé pour sécuriser les liens de communication, parmi ces outils on cite :

c) Les fonctions de hachage

Une fonction de hachage est aussi appelée fonction de hachage à sens unique ou "one-way hash function" en anglais. Ce type de fonction est très utilisé en cryptographie, principalement dans le but de réduire la taille des données à traiter par la fonction de cryptage. En effet, la caractéristique principale d'une fonction de hachage est de produire un haché des données (un condensé de ces données) de taille fixe. Ce condensé est utilisé comme empreinte digitale du message original afin que ce dernier soit identifié de manière univoque.

Les algorithmes de hachage les plus utilisés actuellement sont :

- MD5 [93] (MD signifiant Message Digest) créant une empreinte digitale de 128 bits.
- SHA [94] (Secure Hach Algorithme) créant une empreinte digitale de 160 bits.

d) La signature digitale

C'est un système cryptographique qui repose sur des clés asymétriques pour assurer la non-répudiation de la source. L'émetteur produit une signature digitale et signe le condensé

du message avec sa clé privée. Ce dernier est ensuite envoyé avec les données. Si le destinataire réussi à la déchiffrer avec la clé publique et que le résultat est identique aux données reçus, alors la signature est valide.

e) Le code d'authentification de messages MAC

Le code d'authentification de message MAC (Message Authentication 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. Cette clé est utilisée pour calculer le code MAC par l'émetteur (1). Ce code est par la suite envoyé avec les données (2). Le récepteur calcule à son tour le code MAC avec cette même clé et le compare au code qu'il a reçu (3). S'ils sont bien identiques (4), alors la source est authentique et les données n'ont pas été altérées. La figure I.10 schématise bien cette technique.

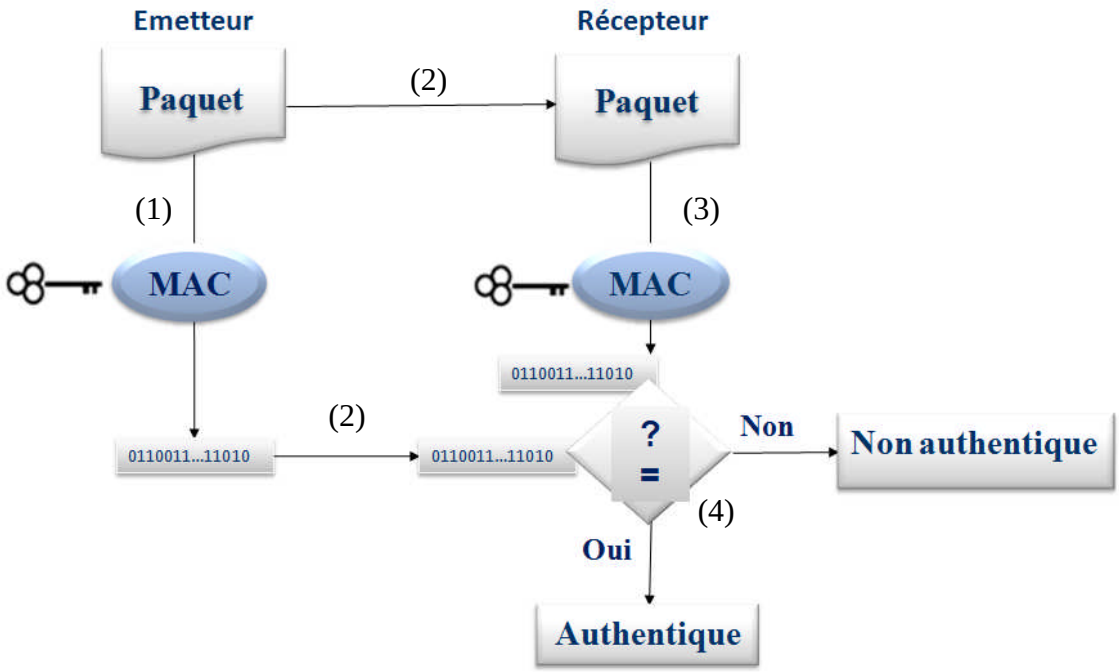


Figure I.10 : Le code d'authentification de message MAC. [51]

4.4.3 Système de Détection d'intrusions (IDS : Intrusion Detection System)

Un système de détection d'intrusion est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (réseau, hôte). Le rôle essentiel d'un IDS réseau est l'analyse et l'interprétation des paquets circulant sur un réseau. Ce système a la capacité de détecter avec une grande précision les attaques internes, et déclenchera une alarme lorsqu'un comportement malveillant se produit.

5 Routage sécurisé dans les réseaux de capteurs sans fil

5.1 Classification des protocoles de routage sécurisés dans les réseaux de capteurs sans fil

Nous illustrons dans la figure I.11 la classification des protocoles de routage sécurisé:

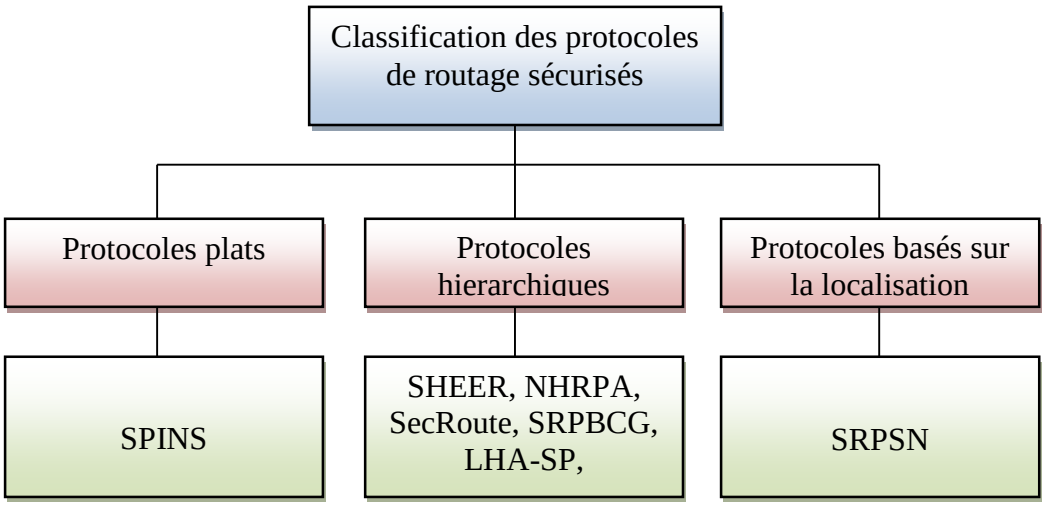


Figure I.11 : Classification des protocoles sécurisés

5.2 Critères d'évaluation des protocoles de routage sécurisé

Nous allons présenter, dans cette section, certaines métriques permettant de mesurer l'efficacité et la robustesse des protocoles de routage sécurisé face aux menaces et contraintes des RCSF. Les performances d'un protocole de routage se mesurent par sa capacité à assurer les tâches de construction et de maintenance de routes et de la transmission d'informations sur le réseau à moindre coûts et à des temps de traitement et de transfert raisonnables. Il est primordial de considérer la consommation d'énergie et l'intégrité des données transmises comme des métriques essentielles pour toute évaluation d'un protocole de routage destinée aux RCSF.

a) Consommation d'énergie

Le nœud capteur doit utiliser son énergie d'une façon optimale pour ses activités de détection, de traitement et de communication. Un protocole de routage doit gérer les périodes d'activité et d'inactivité des capteurs en incluant des modes « en marche » et « en veille » et avoir la notion de temps pour se mettre en veille et se réveiller. Le mode « en veille » permet au nœud capteur d'éteindre son interface radio et d'empêcher une perte d'énergie due aux écoutes actives et en permanence inutiles de son environnement.

b) Temps de traitement

C'est le temps pris par un nœud capteur pour effectuer des opérations de calcul sur les données récoltées ou reçues. Ce temps doit être raisonnable pour ne pas causer des retards de transmission d'informations pour des applications critiques et temps réel.

c) La mobilité des nœuds capteurs

La position des capteurs sur la zone de captage n'est pas toujours fixe. Un nœud capteur peut devenir mobile et changer sa position selon les besoins de l'utilisateur. Des traitements spécifiques pour la maintenance des liens et la mise à jour des informations de routage sont à prévoir lors de la conception d'un protocole de routage.

d) Modes de transmission

Choisir un mode de communication adéquat à la structure et à la topologie du réseau de déploiement.

e) Sécurité des échanges

Consiste l'envoi périodique de paquets de contrôle afin d'éviter des collisions et des pertes de données. Un paquet de contrôle peut contenir le nombre de bits émis, l'adresse ou l'identificateur de destination et des informations sur le routage. Ces paquets sont nécessaires afin d'assurer la disponibilité et l'intégrité des données transmises

5.3 Exemples de protocoles de routage sécurisés

Dans ce qui suit on présente les protocoles de routage sécurisés les plus connues dans les RCSF :

5.3.1 TinySec

TinySec[62] est une librairie de sécurité intégrée au système d'exploitation TinyOS 1.x.L'objectif de cette librairie est de pouvoir détecter les paquets non autorisés lorsqu'ils sont injectés pour la première fois dans le réseau et éviter leur propagation dans le réseau qui amènerait par les communications engendrées, à une perte d'énergie. Pour cela TinySec met en place des mécanismes d'authentification (avec l'utilisation de code MAC), de chiffrement des informations et une protection contre les redondances d'informations. Pour permettre une plus grande liberté d'action, TinySec supporte deux options de sécuritédifférentes :

- TinySec-Auth : la sécurité apportée ne porte que sur une authentification des données.

Les données ne sont pas chiffrées, contrairement au code MAC, calculé à partir de l'entête du paquet pour assurer l'authenticité de l'expéditeur.

- TinySec-AE : la sécurité porte à la fois sur l'authentification mais aussi sur l'encryptage des données. Les données sont chiffrées et envoyées avec un code MAC généré à partir des données chiffrées et de l'entête du paquet (qui contient les informations sur l'expéditeur du message).

Pour l'authentification et l'encryptage des données TinySec utilise un chiffrement par blocs de type CBC-MAC avec l'algorithme de chiffrement SkipJack.De la même manière que pour SNEP, TinySec utilise un vecteur initial pour le premier chiffrement et des chaînes de bits aléatoires ajoutés au message pour empêcher un attaquant d'analyser le trafic par comparaison des paquets.Cependant TinySec n'utilise pas de compteur pour chaque chiffrement, ce qui empêche de garantir la fraîcheur des données et laisse possible les attaques de type rejet de paquets.

5.3.2 SHEER

J. Ibriq et al. [55]ont proposés un protocole de routage hiérarchique sécurisé

(SHEER) qui fournit une communication sécurisée à la couche réseau, Il utilise un mécanisme de diffusion probabiliste et une classification hiérarchique à trois niveaux pour améliorer la performance énergétique du réseau et augmenter sa durée de vie. Pour sécuriser le routage, SHEER utilise un protocole de transmission de clé cryptographique sécurisée et symétrique (HIKES).

5.3.3 NHRPA

Le protocole de routage proposé [56] peut adopter la technologie de routage approprié pour les nœuds en fonction de la distance entre ces derniers et la station de base, la densité de leurs répartitions et leurs énergies résiduelles. NHRPA en termes d'utilisation d'énergie, la latence du paquet et la sécurité en présence d'attaques de nœuds compromis, les résultats montrent que cet algorithme est plus efficace que LEACH pour RCSFs. Il n'utilise aucune

technique de cryptographie, donc les frais généraux sont moins. Mais il ne traite que de l'attaque de nœuds compromis.

5.3.4 SecRoute

Le protocole SecRoute[46] est un protocole de routage hiérarchique sécurisé. Le réseau est organisé en clusters ayant chacun un chef. Le nœud collecteur est supposé connaître cette organisation du réseau, et doit maintenir localement une table contenant une clé secrète de chaque capteur ayant le format illustré par la figure I.12. Cette clé est supposée pré-chargée dans chaque nœud. De plus, chaque cluster doit posséder une clé permettant de sécuriser les échanges intra-cluster. Cette clé doit être connue par le chef de groupe et tous les nœuds du groupe. Ce protocole utilise une architecture à deux niveaux, dans laquelle les chefs agrègent les données des membres puis les transmettent au nœud collecteur. Le protocole SecRoute ne spécifie pas l'algorithme de construction de clusters, et suppose que les clusters ainsi que leurs clés sont établis par un autre protocole, comme LEAP.

Source	*Pre*	*Next*
ID_{Soure}	ID_{pre_2}, ID_{pre_1}	D_{next_1}, D_{next_2}
⋮	⋮	⋮

Figure I.12 : Format de la table de routage dans SecRoute

La table est organisée suivant l'adresse des sources. Le champ Pre (respectivement Next) indique les deux prochains sauts vers la SB (respectivement source) sur le chemin entre la source et la SB.

5.3.5 SRPBCG

Z. Quan et al. [57] a proposé un protocole de routage pour RCSFs appelé protocole de routage sécurisé à base de cluster de gènes (SRPBCG).La sélection de CH est la même que LEACH. L'objectif de ce protocole est de gérer la confiance et la réputation localement et d'authentifier l'identité de nœud avec une charge minimale et un retard temporel.

Le mécanisme d'authentification biologique a été utilisé, qui est une méthode d'authentification très efficace, le «gène» biologique comme clé de chiffrement est un protocole de distribution de clé très sûre et efficace, qui ne nécessitent que peu de mémoire et de charge de communication. Il ne traite que de l'attaque de l'adversaire et les nœuds compromis. Sécurité du protocole est inconsidérément, lors de la formation des clusters et à la transmission du message. La charge de communication est plus élevée dans ce protocole.

5.3.6 SRPSN

Tubaishat et al.[59]Ont proposé le protocole SRPSN (SecuredRouting Protocol for Sensor Network).C'est un protocole de routage hiérarchique pour les réseaux de capteurs pour la sûre-garde de différentes attaques en construisant une route sécurisée de la source à la station de base. Il se base sur le clustering dans lequel les chefs de cluster sont élus et ces derniers collectent des données à partir de ses membres et agrègent les données rassemblées par des procédures de fusion, puis ils les transmettent directement à la station de base. Quand un capteur devient Chef de cluster, ce dernier active son GPS afin de déterminer sa position exacte et de diffuser son identifiant (ID), ainsi que son niveau pour décider sur ses fils les mieux placés pour transmettre les paquets vers la station de base ou vers d'autres destinations. Ce protocole vérifie le critère de stockage, mais dans le cas où le nombre de nœuds dans un

cluster est grand, le coût de calcul des clés de groupes est coûteux, et devient une dépense d'énergie majeure pour les capteurs limités.

5.3.7 LHA-SP

LHA-SP [58] est le premier travail en mettant l'accent sur la sécurisation hétérogène des protocoles hiérarchiques avec un nombre arbitraire de niveaux. Il utilise le système de clé symétrique et il suit l'hypothèse suivante: un adversaire va prendre une certaine période de temps pour compromettre la clé de groupe et si cette quantité de temps dépasse, ça nécessitent la reconfiguration du réseau. Il empêche les intrus (attaquant extérieur) de prendre l'activité, la trempe avec ou injecter des messages dans les réseaux et empêche les écoutes sur la communication entre les nœuds légitimes. L'authentification et la confidentialité est maintenue par une clé partagée entre paires. Il traite des problèmes des nœuds orphelins.

6. Conclusion

Les réseaux de capteurs sans fil connaissent un grand essor grâce à la multitude d'applications qu'ils offrent ainsi que leurs caractéristiques inhérentes telles que leur déploiement aléatoire et de faible coût, leur grande mobilité et aussi, grâce aux récents développements concernant la miniaturisation des composants électroniques (construction des capteurs de quelques millimètres cubes de volume).

Nous avons décrit profondément dans les sections précédentes de ce chapitre les réseaux de capteurs sans fil et les notions de routage et de sécurité.En l'absence de mécanismes de sécurité appropriés, le déploiement des réseaux de capteurs demeure vulnérable à nombreuses attaques. Tandis que la recherche a fait des progrès dans le domaine de la sécurité des réseaux de capteurs, de nombreux défis demeurent sans réponse. Parmi les protocoles existants on s'intéresse particulièrement à la classe des protocoles hiérarchiques. Le protocole LEACH qui manipule des clusters et qui sera étudié dans le chapitre qui suit.

1 Introduction

L'un des protocoles de routage de données les plus simples et les plus couramment employés dans les réseaux de capteurs sans fil est le protocole LEACH (Low Energy Adaptive Clustering Hierarchy, c'est-à-dire «hiérarchie de clusterisation adaptative à faible énergie »). Il s'agit d'un algorithme dynamique qui, par la formation de clusters, met en place une solution de routage simple et efficace des paquets dans le réseau. Néanmoins ce protocole est confronté à plusieurs attaques.

Dans ce chapitre, nous expliquerons l'architecture et l'algorithme détaillé de LEACH et nous discuterons les avantages et inconvénients de ce dernier. Ensuite nous introduirons quelques attaques contre ce protocole et les solutions existantes.

2 Architecture de LEACH :

LEACH a été proposé pour la réduction de la consommation d'énergie. L'idée est de former des clusters de nœuds de capteurs basés sur les zones où il y a un fort signal reçu, puis utiliser des chef de groupe locaux comme passerelle pour atteindre la destination, comme le montre la figure II.1. Cela permet d'économiser de l'énergie car les transmissions ne sont effectuées que par les chefs de groupes plutôt que par tous les nœuds de capteurs [50]

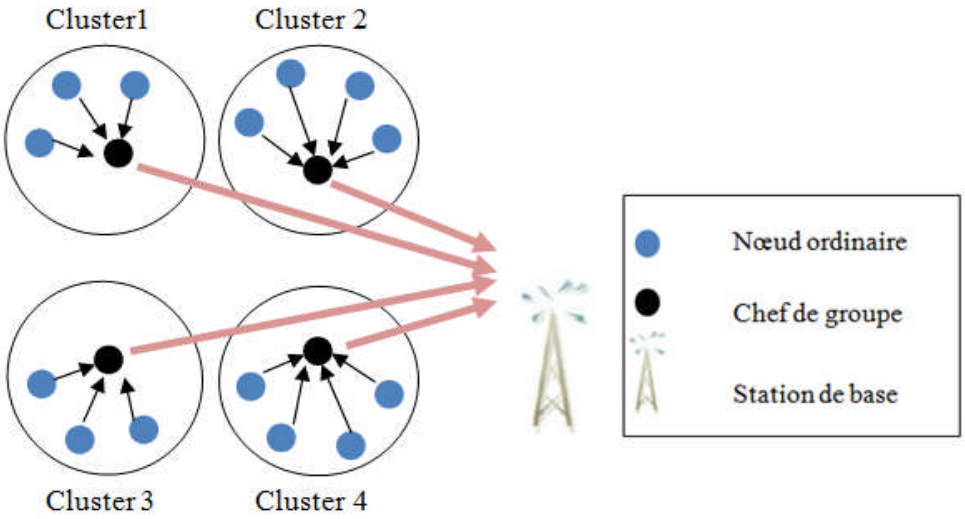


Figure II.1 : Architecture de LEACH

3 Protocoles MAC utilisés par LEACH

Pendant son fonctionnement, le protocole LEACH appelle certains schémas du protocole MAC qui seront détaillés dans cette section pour mieux comprendre son déroulement. Les nœuds doivent avoir une certaine capacité de calcul pour supporter différents protocoles MAC. Comme les RCSF ont des caractéristiques distinctes de tout autre type de réseaux sans fil, les protocoles MAC conçus pour ces derniers ne sont pas toujours applicables dans les RCSF. Deux versions des protocoles MAC pour l'accès au media sont alors proposées pour les RCSF : l'accès aléatoire et l'allocation fixe.

3.1. Accès aléatoire

Les schémas à accès aléatoire sont à base de contention. Dans ces derniers, les nœuds qui possèdent des données à transmettre doivent essayer d'obtenir l'autorisation pour l'accès au media tout en réduisant les collisions avec les transmissions des données des autres nœuds. Le schéma d'accès multiple avec surveillance de porteuse CSMA (Carrier Sense Multiple Access) sur lequel se base le protocole LEACH est l'un des schémas d'accès aléatoire [38].
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 un délai aléatoire que chaque nœud attende avant de retransmettre ses données, ce qui réduit la probabilité d'une autre collision.

3.2. Allocation fixe

Les schémas à allocation fixe permettent d'allouer pour chaque nœud le media de transmission suivant des intervalles de temps (schéma TDMA) ou un schéma de codage particulier (schéma CDMA).Étant donné que chaque nœud est attribué en exclusivité à un intervalle, il n'y a presque pas de collisions entre les données. Toutefois, les schémas à allocation fixe s'avèrent inefficaces lorsque tous les nœuds n'ont pas de données à transmettre. En effet, ces intervalles sont affectés à des nœuds qui n'ont pas besoin de les utiliser [39].

- TDMA

  Dans cette technique, la bande passante est utilisée par tous les utilisateurs mais la division se fait sur l'axe de temps. Chaque utilisateur envoie sur un intervalle de temps et en utilisant toute la bande passante. Les données envoyées par chaque utilisateur sont groupées en rafales pour être envoyés sur des intervalles de temps appelés slots. Le canal se comporte donc comme la succession des slots remplis par des rafales venant des différents utilisateurs. Si la durée d'un slot est T_s le canal peut contenir n slots ; on appelle trame l'ensemble des n slots du canal. La durée d'une trame est alors de $T_f = n \times T_s$. La figure II.2 représente la technique d'accès TDMA. Le récepteur doit identifier chaque paquet dans un slot afin de lire les informations qui lui sont destinées. Ceci nécessite des informations d'identification du début d'un paquet ainsi qu'une synchronisation [60].

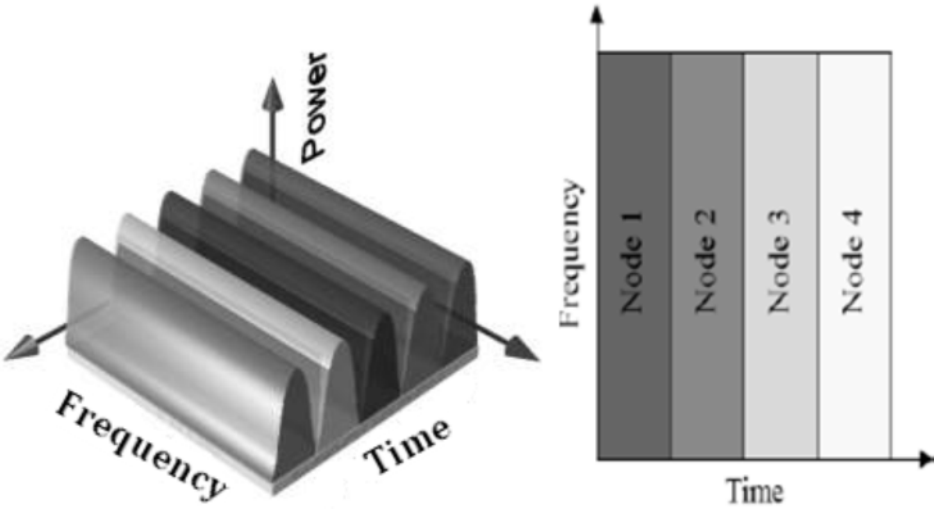


Figure II.2 : Technique d'accès TDMA [80].

- CDMA

Dans cette technique [37], chaque utilisateur transmet ses informations sur le canal continûment et en utilisant toute la bande passante. Ceci veut dire qu'il y a interférence entre les différents utilisateurs, mais chaque utilisateur envoie sa propre signature avec ses informations. Cette signature est appelée code (désigné par p_i) et elle est combinée avec les informations utiles avant de tout transmettre. Un émetteur choisit son propre code d'un ensemble des codes caractérisés par les propriétés suivantes :

- Chaque code doit être facilement distingué de sa répétition dans le temps.
- Chaque code doit être facilement distingué des autres codes utilisés.
- Les différents codes sont pseudo-orthogonaux et donc $p_i \times p_j \approx 0 \forall\ i \neq j$.

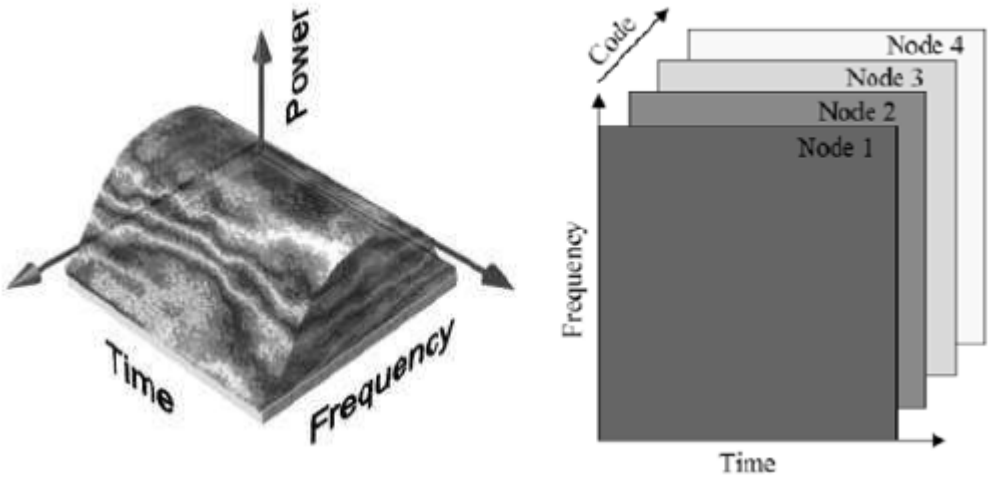


Figure II.3 : Technique d'accès CDMA [80].

4 Algorithme détaillé de LEACH :

L'algorithme se déroule en « *tours* » qui ont approximativement le même intervalle de temps déterminé au préalable. Chaque round est constitué de deux phases : la phase « set-up »

et la phase « steady-state ». Dans la première phase, les CHs sont sélectionnés et les clusters sont formés, et dans la seconde phase, le transfert de données vers la station de base aura lieu.

4.1. Phase d'initialisation (set-up phase)

Comme l'indique la figure II.4, la phase d'initialisation est composée de 3 sous phases: d'annonce, d'organisation des groupes et enfin d'ordonnancement.

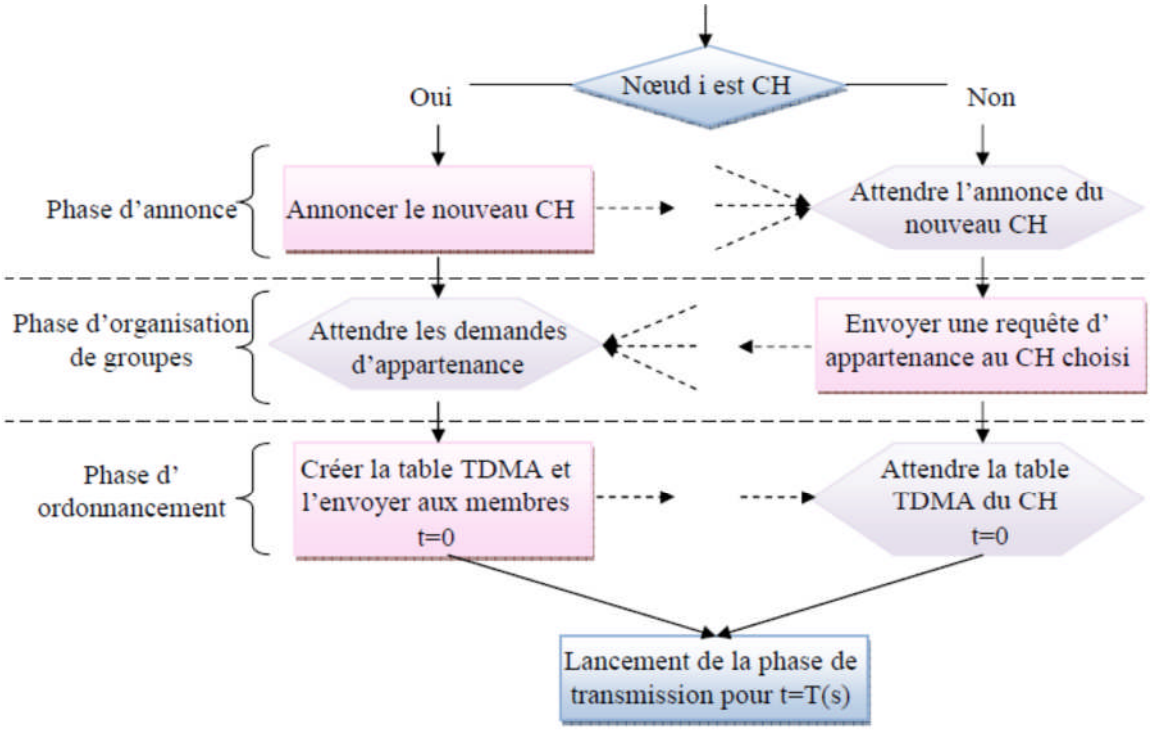


Figure II.4 : Opérations de l'étape d'initialisation de LEACH.

4.1.1. Phase d'annonce

Dans cette phase, le processus d'élection des chefs de groupes est déclenché pour choisir les futurs chefs de groupes. Ainsi, une fraction prédéterminée de nœuds s'élisent comme chefs de groupes selon le schéma d'exécution suivant : durant une période, un nœud n choisit un nombre aléatoire nb dont la valeur est comprise entre 0 et 1 ($0 < nb < 1$). Si nb est inférieure à une valeur seuil T(n) alors le nœud n deviendra chef de groupe durant la période courante, sinon le nœud n devrait rejoindre le CH le plus proche dans son voisinage.

Le seuil est défini comme suit :

$$T(n) = \begin{cases} \dfrac{P}{1 - P * (r \bmod \frac{1}{P})} & \text{Si } n \in G \\ 0 & \text{Sinon} \end{cases}$$

- P est le pourcentage souhaité de CH ;
- r est le tour courant ;
- G est l'ensemble des nœuds qui n'ont pas été CH lors des (1/p) tours précédents.

4.1.2. Phase d'organisation des groupes

Après que chaque nœud ait choisi quel groupe rejoindre, le CH doit être informé des nœuds composant son groupe. Chaque nœud non-CH transmet une requête de ralliement vers le CH choisi en utilisant également un protocole CSMA MAC. Ce message ou requête de ralliement se compose de l'ID du nœud, de l'ID du CH et d'une entête. Durant cette phase, tous les CH doivent maintenir leurs récepteurs allumés.

4.1.3. Phase d'ordonnancement

Après avoir reçu les requêtes de tous les nœuds l'ayant rejoint, le CH leur alloue chacun un tour dans un ordonnancement TDMA (Time Division Multiple Access) en se basant sur le nombre de nœuds. Et ce n'est que pendant le temps qui leur est alloué que les nœuds peuvent transmettre leurs données vers le CH.

4.2. Phase de transmission (steady-state phase)

Cette phase est plus longue que la phase précédente, et permet la collecte de données captées. En utilisant l'ordonnanceur TDMA, les membres émettent leurs données captées pendant leurs propres slots. Cela leur permet d'éteindre leurs interfaces de communication en dehors de leurs slots afin d'économiser leur énergie. Ces données sont ensuite agrégées par les CH qui les fusionnent et les compressent puis ils envoient le résultat final à la station de base. Après un intervalle de temps prédéterminé, le réseau va passer à un nouveau round. Ce processus est répété jusqu'à ce que tous les nœuds du réseau seront élus CH. une fois que tous les nœuds seront élus CH, le round est réinitialisé à 0.

5 Avantages et inconvénients de LEACH :

Bien que LEACH économise la consommation d'énergie comparé à la transmission directe, grâce à l'agrégation de données, un nombre d'inconvénients restent plus ou moins apparents. Dans ce qui suit, nous citerons quelques avantages et inconvénients du protocole LEACH.

5.1.Avantages

Parmi les avantages du protocole LEACH [4][5] :

- Il fournit la scalabilité (évolutivité) dans le réseau en limitant la plupart des communications à l'intérieur des différents groupes (clusters) du réseau.
- Les cluster-heads (CH) agrègent ou fusionnent les informations rassemblées par les nœuds capteurs, ce qui aide à limiter le trafic produit dans le réseau. Ainsi, un réseau à grande échelle sans surcharge de trafic peut être déployé et une meilleure topologie préservant l'énergie peut être réalisée en comparaison à la topologie plate.
- La propriété de distributivité du rôle de CH entre les membres d'un cluster.
- Il ne requiert pas d'informations sur la localisation des nœuds capteurs dans le réseau afin de former les groupes.
- L'utilisation des techniques TDMA/CDMA permet d'avoir une hiérarchie et de réaliser des clusterings sur plusieurs niveaux. Ces derniers permettent d'économiser l'avantage d'énergie.
- La consommation d'énergie est partagée sur l'ensemble des nœuds prolongeant ainsi la durée de vie du réseau.

5.2.Inconvénients

Le protocole LEACH présente les inconvénients suivants [6][7] :

- Il s'appuie significativement sur les CH plutôt que sur les nœuds membres du cluster pour communiquer avec la station de base. De ce fait, il encourt des problèmes de robustesse comme la défaillance des CH.
- Les CH ne sont pas uniformément distribués dans le cluster, ce qui signifie que les CH peuvent se situer sur les bords du cluster. Par conséquent, certains nœuds n'auront pas de CH dans leurs voisinages.
- Il n'y a pas de communication intergroupe dans le réseau car les CH communiquent directement avec la station de base. Ce processus nécessite une grande gamme de puissance de transmission dans le réseau. C'est pour cela que LEACH n'est pas le mieux adapté pour les réseaux de grande envergure qui requièrent une communication à un seul saut avec la station de base.
- Les CH les plus éloignés de la station de base meurent rapidement par rapport à ceux qui sont proches de la station.
- On pourra ne pas avoir des CH durant un round si les nombres aléatoires générés par tous les nœuds du réseau sont supérieurs à la probabilité T(n).
- Aucune suggestion n’est faite à propos du temps de réélection des CH (temps des itérations).
- LEACH ne fournit pas de clarté sur la position des nœuds capteurs et le nombre de CH dans le réseau [35].
- Le protocole LEACH n’est pas sécurisé. Aucun mécanisme de sécurité n’est intégré dans ce protocole. Ainsi, il est très vulnérable même aux simples attaques. Donc, un attaquant peut facilement monopoliser le réseau et induit à son disfonctionnement.

6 Les variantes de LEACH :

Dans ce qui suit nous allons présenter quelques variantes du protocole de routage LEACH

6.1.TL-LEACH

Différemment de LEACH, ce protocole fonctionne en deux niveaux d’hiérarchie. Au lieu d’être transférées directement à la station de base, les données agrégées par chefs de groupes seront collectées par un chef de groupe situé entre les chefs de groupes et la station de base comme le montre la figure II.5. Ce protocole à l’avantage de réduire la consommation d’énergie [64].

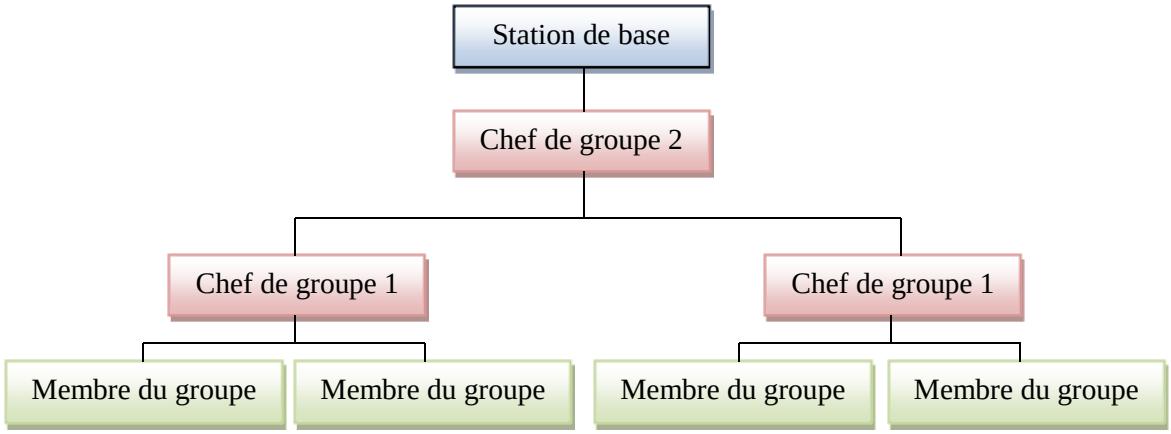


Figure II.5 : Le protocole TL-LEACH.

6.2.E- LEACH

Dans ce protocole, initialement tous les nœuds ont le même niveau d'énergie et la même probabilité de devenir CH. Après le premier tour, le niveau d'énergie de tous les nœuds change. Ainsi l'énergie résiduelle de chaque nœud est utilisée pour désigner le nœud CH. Les nœuds ayant les plus grands nivaux d'énergie résiduelle sont favorisés. E-LEACH augmente la durée de vie du réseau en balançant l'énergie restante entre tous les nœuds du réseau comme le montre la figure II.6.

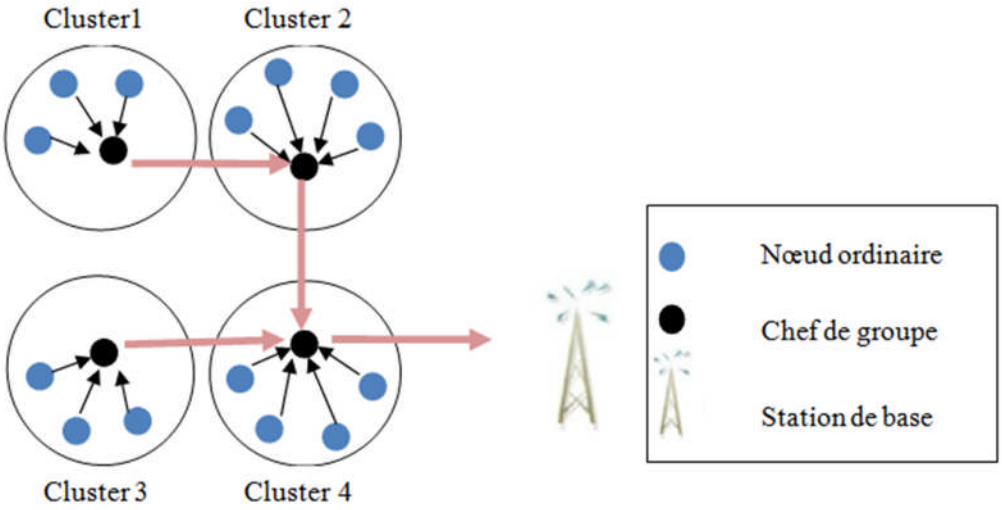


Figure II.6 : Le protocole E-LEACH.

6.3.LEACH-C

C'est un algorithme itératif [66], dans lequel la structure des clusters est calculée au niveau de la station de base en utilisant la méthode d'optimisation du "Circuit simulé" [36]. Cependant, la station de base affecte dans chaque itération des rôles pour les différents nœuds du réseau (CH ou nœud ordinaire). Ensuite, le fonctionnement continue de la même manière que pour LEACH.

6.4.LEACH-F

Comme LEACH-C, ce protocole utilise une approche centralisée pour la formation de cluster. Une fois le processus de formation de clusters est fini, il n'y aura pas de phase de re-clustering dans le prochain tour. Les clusters sont fixés et la rotation des CH sera faite uniquement dans leurs clusters, c'est-à-dire, le nouveau CH sera un nœud qui appartient au même cluster.

6.5.LEACH-A

Parmis les inconvénients de LEACH, les nœuds chefs de groupes consomment plus d'énergie que les autre nœud normaux . LEACH-A est un protocole utilisé pour reduire la mort des nœuds et l'extension d'intervalle de temps avant la mort du premier nœud (appelé periode de stabilité) . Soit n le nombre de nœuds dans le reseau et m la fraction de n qui ont un niveau d'énergie plus grand que les autres, ces nœuds sont appelés des nœuds CGA (utilisés comme passerelles ou chefs de groupes) [72][73] comme l'illustre la figure II.7 .

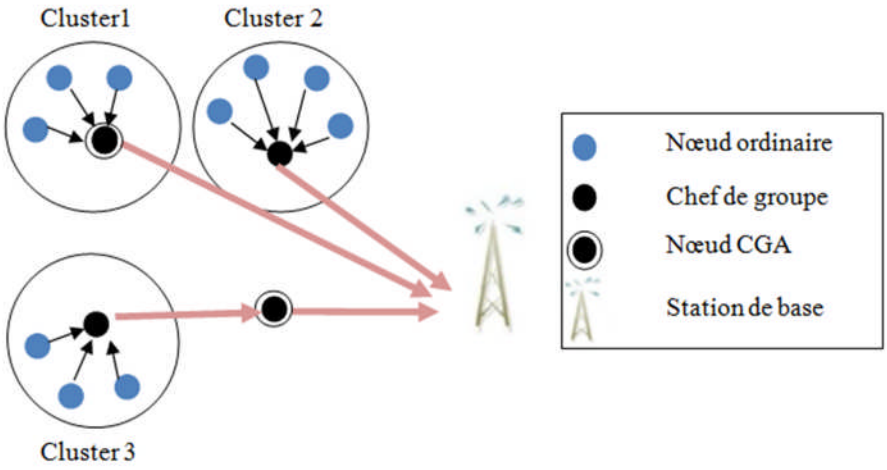


Figure II.7 : Le protocole LEACH-A

6.6.V- LEACH

Dans le protocole LEACH, les CH consomment plus d'énergie par rapport aux nœuds normaux dans l'envoi de données agrégées à la station de base (situé loin). Par conséquent, le nœud principal du cluster meurt tôt et l'ensemble du cluster deviendra inutile, et cela entraîne la perte de données. V-LEACH améliore cet inconvénient en introduisant un vice CH dans chaque groupe qui prend le rôle de chef de cluster lorsque celui-ci meurt. De cette façon, ce protocole réduit les frais généraux de sélection d'une nouvelle tête de grappe à chaque fois qu'une tête de cluster meurt et les données atteindront toujours la station de base comme on le voit sur la figure II.8. Ainsi la durée de vie du réseau augmente [65].

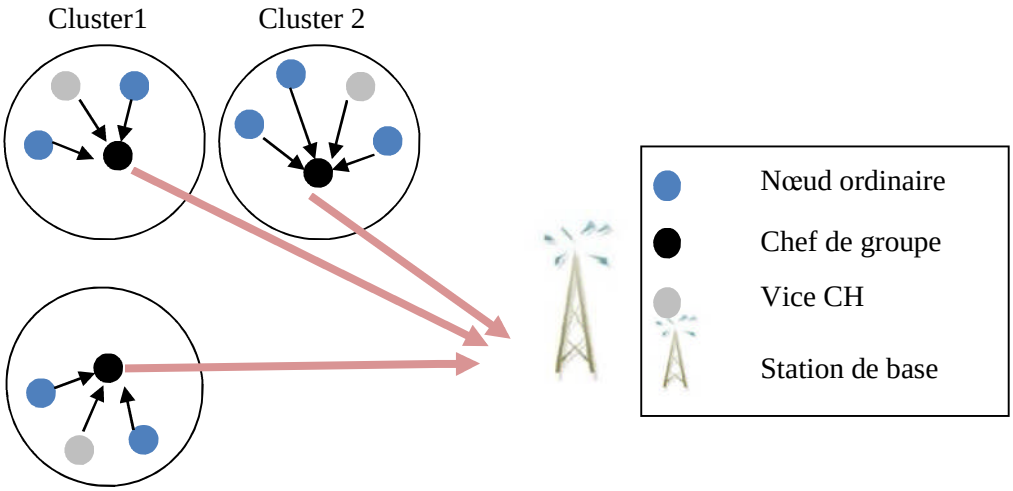


Figure II.8 : Le protocole V-LEACH

6.7.M- LEACH

Ce protocole assure la mobilité de tous les nœuds (CH ou nœuds ordinaires) dans les phases d'initialisation et de transmission. Les nœuds sont homogènes et l'emplacement de chaque nœud est calculé par GPS. Les nœuds moins mobiles et ayant un niveau d'énergie élevée sont désignées comme chefs de groupes. Et le rôle des CH est diffusé à tous les nœuds dans sa plage de transmission [70].

6.8.LEACH-B

A chaque tour, après la première sélection de tête de grappe selon le protocole LEACH, une seconde sélection est introduite pour modifier le nombre de têtes de clusters en contrepartie de l'énergie résiduelle du nœud. En conséquence, le nombre de têtes de cluster est constant et prés optimale par tour. Ce protocole améliore et équilibre la consommation d'énergie du système et prolonge la durée de vie du réseau [67] [68].

6.9.MH- LEACH

Dans LEACH le nœud CH envoi les données directement à la station de base sans prendre en compte la distance entre eux, cela va causer une grande consommation d'énergie si la station de base est située à une grande distance. Plus le diamètre du réseau augmente, plus la distance entre la station de base et les nœuds CH augmente. Pour augmenter l'économie d'énergie, les communications multi-sauts sont introduites. Initialement, les nœuds membres d'un cluster envoient des données à leurs CH. Ensuite, ces derniers transfèrent les données aux autres chefs de groupes jusqu'à atteindre la station de base comme le montre la figure II.9 [69].

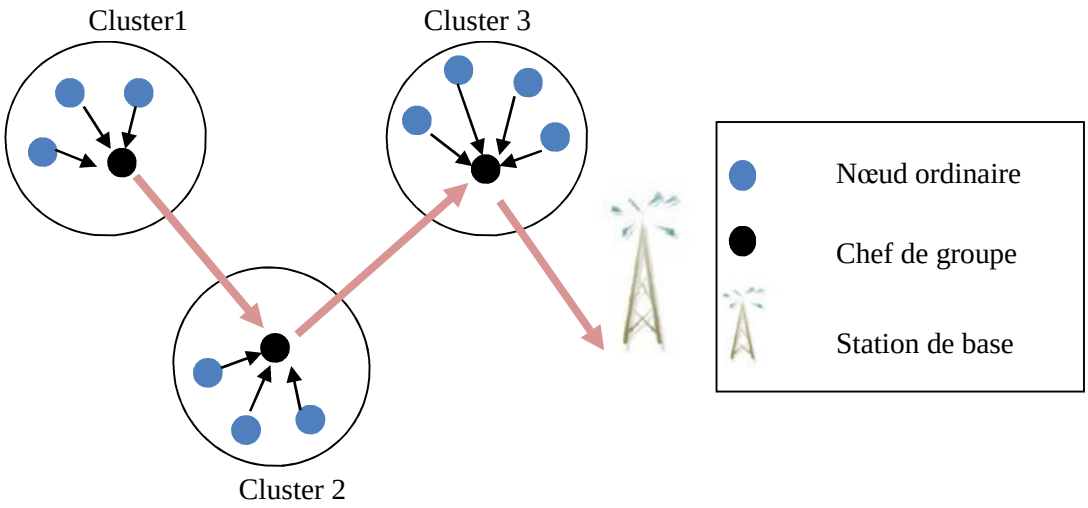


Figure II.9: Le protocole MH-LEACH.

6.10. I- LEACH

La détection des nœuds jumeaux et l'attribution des nœuds Sous-CH sont les deux fonctions réalisées par I-LEACH. Le déploiement aléatoire des nœuds entraîne une forte probabilité que deux nœuds soient situé très près l'un de l'autre, ces nœuds sont appelés nœuds jumeaux (ou Twin nodes en Anglais). Il est nécessaire de garder en veille l'un des nœuds jusqu'à ce que l'énergie de l'autre soit épuisée. Par conséquent, I-LEACH a une distribution uniforme des CH afin qu'il n'y ait pas de manque d'énergie lorsque des transmissions à grande distance auront lieu. Ce protocole utilise l'approche de seuil pour la gestion du nombre de nœuds du cluster pour chaque CH dans le réseau [63].

6.11. Cell- LEACH

Dans Cell-LEACH, le réseau est divisé en plusieurs groupes où chaque groupe est en outre divisé en 7 sections appelées cellules. Plusieurs capteurs sont inclus dans chaque cellule à partir de laquelle un nœud est sélectionné comme tête de cellule. Une fois les cellules et les groupes sont formés, cette répartition est maintenue. Chaque nœud de la cellule envoie des données à la tête de la cellule à l'instant attribuée par TDMA. Une fonction d'agrégation de données est effectuée par les chefs de cellules et les données traitées sont envoyées aux chefs de cluster. Les CH remplissent la même fonction que les chefs de cellules et transfèrent les données à la station de base comme l'illustre la figure II.10, Après la première manche, la tête de la cellule et la tête du cluster seront déterminés au hasard [71].

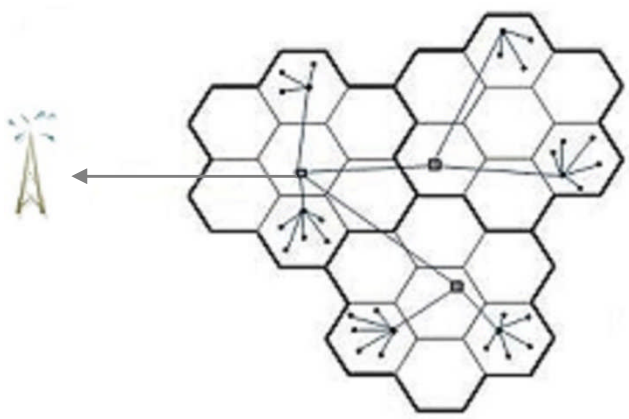


Figure II.10 : Le protocole Cell-LEACH.

7 Les attaques contre LEACH :

En raison de la transmission des données qui est en diffusion, les réseaux de capteurs comme les autres réseaux sans fil sont sensibles aux différentes attaques qui menacent la sécurité du réseau. En outre, les réseaux de capteurs ont une caractéristique spécifique qui les rend plus vulnérables que les autres technologies sans fil classiques. Comme la plupart des protocoles de routage dans les RCSF, LEACH est vulnérable à un certain nombre d'attaques de sécurité, Toutefois étant un protocole à base de clusters nous comptons entièrement sur le CH pour la tâche d'agrégation des données et de routage. Par conséquent les attaques qui visent les CH sont préjudiciables. Si un intrus parvient à devenir un CH, il peut provoquer des crises, perturbant ainsi le fonctionnement du réseau. Nous présentons dans la suite les principaux types d'attaques contre le protocole de routage LEACH.

7.1.Attaque du Trou Noir "black hole"

Dans l'attaque du trou noir [75], l'intrus (nœud malveillant, qui s'introduit illégitimement) modifie les informations de routage afin d'obliger le passage des informations par lui-même. Ensuite il crée un trou noir qui aspire toutes les données qui lui sont transmises comme le montre la figure II.11. Le nœud malveillant peut aussi se placer dans un endroit de routage stratégique et supprimer tous les messages qu'il devrait retransmettre, causant la mise hors service de tout le réseau, et la suspension du service de routage du réseau dans les routes qui passent par le nœud intrus.

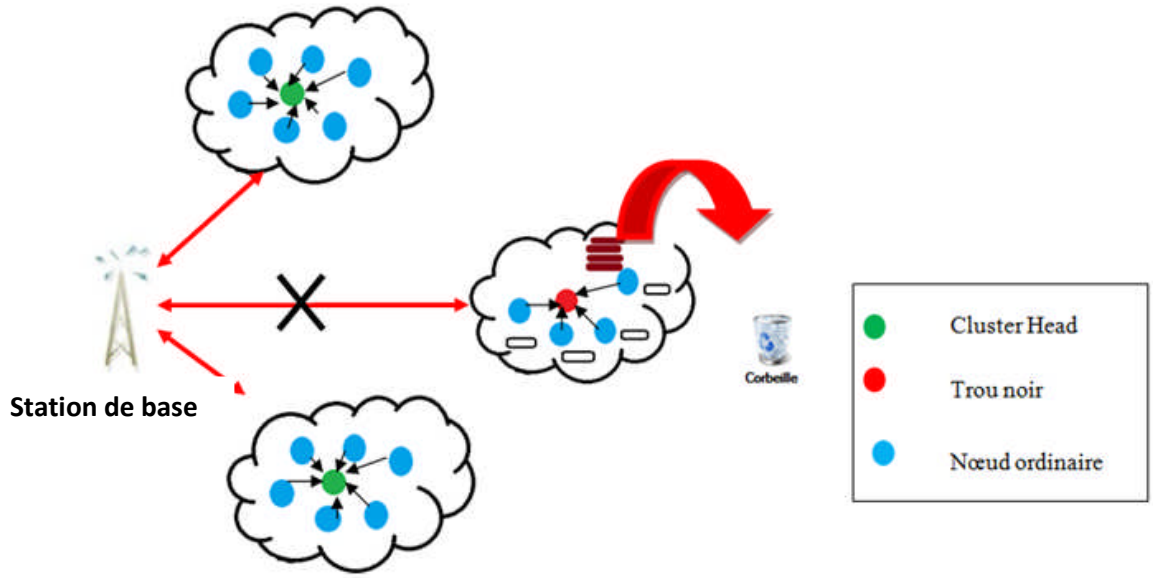


Figure II.11 : Attaque trou noir dans LEACH.

7.2. Attaque du Trou Gris "greyhole"

L'attaque du trou gris [76] est une variante améliorée de l'attaque du trou noir. Tout comme le principe du trou noir, l'attaque trou gris procède à la modification des tables de routage des nœuds du réseau par l'insertion d'un nouveau nœud ou la compromission d'un nœud du réseau, mais à la différence que les informations récupérées par l'attaquant du trou gris ne seront pas toutes détruites et quelques informations non critiques seraient acheminées correctement comme on le voit dans la figure II.12. Ce comportement semble normal aux autres nœuds du réseau d'où sa détection est rendue plus difficile.

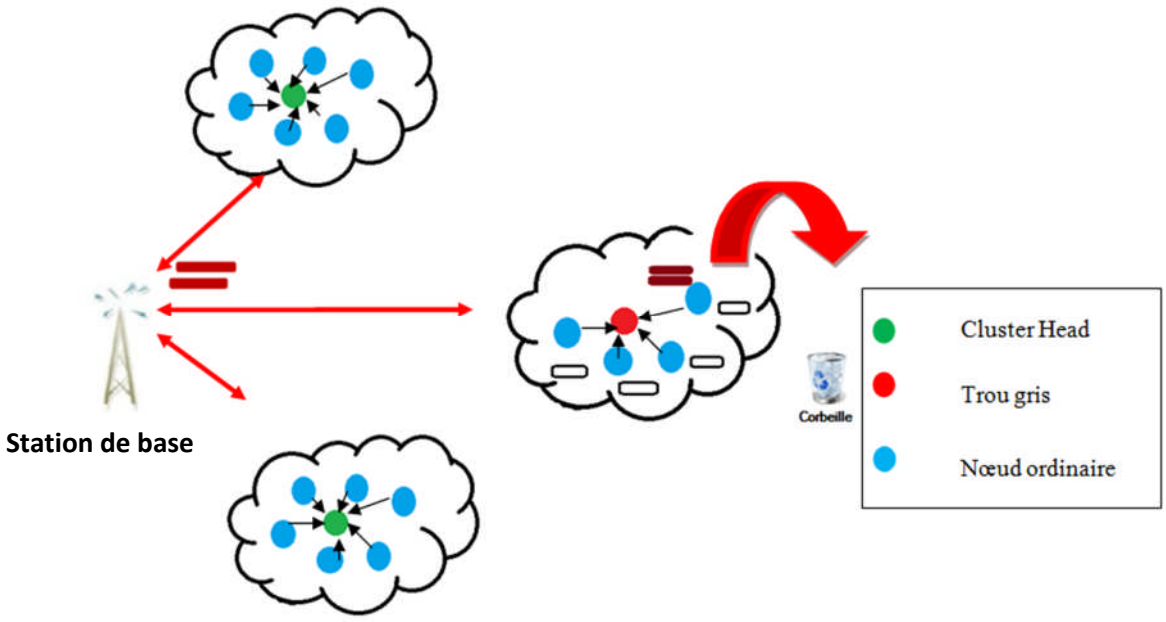


Figure II.12 : Attaque trou gris dans LEACH.

7.3. L’attaque du trou de base " SinkHole attack" :

L’attaquant achemine tout le trafic vers lui afin de contrôler la plupart des données circulant dans le réseau. L'attaquant apparaît ainsi aux autres nœuds comme étant la station de base en émettant un signal plus fort que celui de la station de base originale. Du coup toutes les informations qui y transitent pourront être récupérées par l’attaquant. 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 [74].

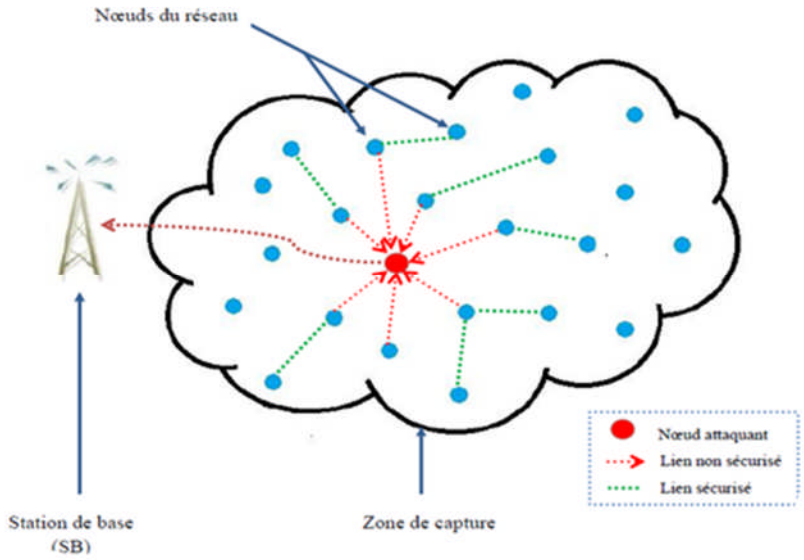


Figure II.13 : Attaque du trou de base

7.4. L’attaque Jamming

Dans cette attaque, le nœud malveillant essaye d’interférer avec la fréquence radio utilisée par les nœuds capteurs dans le réseau. La source de brouillage peut être assez puissante pour perturber l'ensemble du réseau. Le nœud malveillant peut lancer des attaques de brouillage stratégiques en ciblant des zones sensibles du réseau (station de base ou chef de cluster) sans attirer les attentions (signal de brouillage qui respecte les normes du réseau*).*

7.5. L’attaque Hello Flood

L’inondation par des messages HELLO. Le nœud malveillant et lors de l’étape de formation des clusters diffuse pour s'annoncer à ses voisins des messages Hello avec une grande puissance de transmission suffisante qui pourrait convaincre chaque nœud du réseau que l'adversaire est son voisin, et un nœud recevant un tel paquet peut supposer que c'est à l'intérieur du réseau. Apres l’élection d’un CH en se basant sur la force du signale reçu, L'adversaire peut choisir de faire un rejet sélectif de données qui l’atteignent réellement, ou n’achemine aucune donnée et le réseau va être effectivement désactivée.

7.6. Spoofed Cluster Head

L'attaquant peut présenter un nœud malveillant ayant une grande puissance du signale par l'utilisation de l'attaque hello flood qui va par conséquent être choisi à la tête du cluster. Cela lui permettra de devenir le chef de groupe pour une grande partie du réseau ou pour le réseau tout entier (Sinkhole). En tant que chef de cluster l'attaquant peut choisir de passer tous les paquets ou de les négliger. Il aurait aussi la capacité à modifier les slots de temps TDMA qui malicieusement va provoquer des collisions entre les transmissions par différents nœuds capteurs.

8 Variantes sécurisé de LEACH

Maintenant que nous avons présenté les différentes attaques contre le protocole de routage LEACH, nous allons présenter certains protocoles sécurisés qui assure la protection du protocole LEACH contre d'éventuelles attaques.

8.1. Sec-LEACH

Sec-LEACH fournit une solution efficace pour assurer la sécurité des communications dans LEACH, il utilise la pré distribution de clés et μTESLA[97] pour sécuriser les réseaux de capteurs hiérarchiques avec la formation de clusters dynamique, Sec-LEACH applique la distribution de clé aléatoire sur LEACH et introduit une clé symétrique et une chaîne de hachage à sens unique de manière à assurer la confidentialité et la fraîcheur des données. Sec-LEACH assure l'authentification, l'intégrité, la confidentialité et la fraîcheur des données aux communications [79].

8.2. S-LEACH

Le protocole S-LEACH [77] est une extension sécurisée du protocole LEACH. S-LEACH assure la sécurité dans LEACH en utilisant SPINS [96] (Security Protocol for Sensor Network) et les méthodes à clés symétriques et MAC (Message Authentication Code). S-LEACH protège contre l'attaque trou de base et HELLO flooding. Il empêche l'intrus d'envoyer des données erronées au CH et l'empêche de transmettre ces faux messages à la station de base. La solution est destinée à protéger seulement le réseau contre les attaques externes.

8.3. F-LEACH

LB. Oliveria et al. Ont proposé le protocole sécurisé F-LEACH [78], c'est un protocole qui permet de sécuriser la communication entre deux nœuds dans le réseau de capteurs. Il utilise un système de pré-distribution de clé aléatoire avec la cryptographie à clé symétrique pour améliorer la sécurité dans le protocole de base LEACH. F-LEACH permet d'assurer les services de sécurité les plus importants : l'authentification, l'intégrité, la confidentialité et fraîcheur de données entre deux nœuds qui communiquent. Mais il est vulnérable à l'attaque de capture des nœuds capteurs.

9 Conclusion

Dans ce chapitre, nous avons présenté le protocole de routage hiérarchique LEACH dont l'objectif principal est le prolongement de la durée de la vie du réseau ainsi que la gestion efficace de la consommation énergétique.
Nous avons vu que LEACH est un protocole à base de cluster, en s'appuyant fondamentalement sur les CH pour l'agrégation des données et le routage. Cependant, puisqu'ils sont considérés comme les points les plus vulnérables dans le réseau, les attaques peuvent les viser en brouillant seulement les liens de communication entre ces CH et la station de base. Pour cela, nous proposerons dans le prochain chapitre une alternative sécurisée du protocole LEACH qui permet de pallier aux problèmes de sécurité les plus importants.

1. Introduction

Les réseaux de capteurs ont vu le jour à la fin des années 90. Leur évolution ne cesse de s'accroitre. La sécurité reste l'un des principaux objectifs des protocoles conçus pour ce type de réseaux.

Le routage est l'un des axes les plus sensibles et les plus importants dans les RCSF car il constitue la colonne vertébrale du réseau par le bon fonctionnement de ces mécanismes. Les protocoles de routage hiérarchiques se présentent comme l'une des solutions les plus efficaces face aux contraintes du routage (la redondance de donnée, l'absence d'adressage, la consommation d'énergie…etc.) qui sont considérées comme des problèmes de base pour le routage.

On a choisi de sécurisé le protocole MH-LEACH PSM, une variante de LEACH pour ses nombreux avantages, dans ce chapitre nous allons proposer une solution de sécurisation pour ce protocole, qui permet de protéger le réseau contre les attaques de type trou noir et de type trou de base.

2. Problématique

LEACH est un protocole de routage hiérarchique à base de clusters. Tous les nœuds d'un cluster sont chargés de collecter des données environnementales et de les router à leurs chefs de groupe (CH). Ces derniers recueillent ainsi toutes les données de leurs clusters, ensuite ils les agrègent et les routent vers la station de base. De ce fait, les CH et la station de base font des cibles de choix pour les attaquants, dans le but de créer un disfonctionnement sur les réseaux hiérarchiques. Plusieurs menaces contre les RCSF ont été décrites dans les chapitres précédents comme l'attaque trou noir, l'attaque trou de base, l'attaque Jamming etc...

Dans les réseaux à topologie hiérarchique, l'attaque est plus dangereuse si l'attaquant devient Cluster-Head ou station de base. Pour cela, nous nous sommes intéressés à deux attaques d'abord l'attaque du trou noir, qui est un nœud attaquant, qui essaye par divers moyens de se positionner dans le réseau comme étant un CH ensuite l'attaque trou de base dans ce cas l'attaquant apparaît aux autres nœuds comme étant la station de base en émettant un signal plus fort que celui de la station de base originale.

Pour préserver les bonnes opérations du réseau, la sécurité doit être considérée comme une composante essentielle du mécanisme de routage. Nous allons donc, proposer un mécanisme pour la détection des attaques trous noirs et trous de base. Ainsi, nous obtenons un nouveau protocole de routage sécurisé.

La sécurisation doit être en mesure de garantir les services suivants :

- **Offrir un niveau de sécurité acceptable**: Prendre en charge les services de sécurité requis par le protocole.
- **Faire face aux attaques** : Pallier aux principales attaques étudiées dans le chapitre précédent.
- **Maintenir les performances du réseau**: Ne pas dégrader les performances du réseau après la sécurisation.

3. Motivations

Dans notre travail, on a choisi l'approche de routage hiérarchique pour ses nombreux avantages (agrégation de données, la répartition des charges par les tours de rôle des nœuds-chef…etc.). LEACH est considéré comme une référence des protocoles hiérarchiques du fait

qu'il est le premier protocole conçu pour ce type d'approche de routage. On a vu dans le [chapitre 2] une étude de ce protocole, son fonctionnement, ses avantages et ses inconvénients devant lesquelles nous nous sommes arrêtés pour présenter quelques-uns :

- Le risque d'absence des CH (nœuds-chef) si le nombre aléatoire générés par tous les nœuds est supérieur à la probabilité P_i, comme il est possible d'avoir un nombre supérieur à celle désirée auparavant si les nombre générés par plusieurs nœuds sont inférieurs à P_i, ce qui provoque une large consommation d'énergie par les capteurs qui jouent le rôle de CH.
- L'instabilité du réseau suite au changement de la topologie à chaque nouveau round, cette restructuration implique une dissipation énorme d'énergie.
- La rotation du rôle des CH se fait sans aucune contrainte, elle est totalement aléatoire (puisqu'elle dépend du nombre aléatoire généré par les nœuds et la valeur de probabilité P_i) donc il est possible d'avoir un CH avec une faible capacité énergétique ce qui provoque le dysfonctionnement du réseau.

Tous ces inconvénients et bien d'autres nous laissent à repenser à utiliser un protocole hiérarchique différent de LEACH (sous forme d'une variante) contenant des solutions efficaces aux insuffisances du protocole LEACH tout en étant efficace en terme d'énergie pour cela on a opté pour le protocole MH-LEACH PSM qui est une variante de LEACH.

4. Présentation de la variante MH-LEACH PSM

Le protocole utilisé est un protocole hybride, centralisé dans le premier round ou la station de base désigne aléatoirement les Clusters Head selon le nombre désiré, et reparti dans les rounds qui suivent le premier jusqu'à ce que tous les nœuds passent par le rôle du CH.

La sélection du CH dans ce protocole est basée principalement sur l'énergie résiduelle du nœud capteur. Cette sélection est faite d'une manière aléatoire puisqu'on considère que tous les nœuds capteurs sont homogènes donc ils possèdent une même énergie résiduelle initiale lors du déploiement, et elle sera presque la même après le tournement du rôle du CH.

L'élection du prochain CH pour le prochain round est basée sur l'énergie résiduelle envoyée par les nœuds avec la donnée captée dans leurs slots, le CH choisi le nœud ayant le plus grand taux d'énergie résiduelle pour jouer le rôle du CH dans le prochain round, cela va réduire les messages échangés pendant la phase d'initialisation (formation des grappes) au niveau de chaque nœud dans le cas d'une reformation des clusters.

Fonctionnement du protocole

1- round =0

a) la phase initialisation

- Le nœud puits initialise le round à 0 et choisi aléatoirement les Clusters Head selon le nombre des clusters désiré (généralement 10% est le pourcentage des Clusters Head sur le nombre totale des nœuds), après il lance le message de déclenchement du nouveau round 0 correspond à la formation des grappes contenant les adresses des CH sélectionnés.
- Chaque nœud reçoit le message du déclenchement du round, il vérifie s'il est élu comme CH, et si c'est le cas, il invite les nœuds à joindre son cluster.
- Le nœud qui reçoit des invitations pour rejoindre les Clusters, il choisit le Cluster Head le plus proche (pour implémenter l'amplification du signal) et lui envoie une demande d'adhésion.

- Le Cluster Head reçoit les demandes d'adhésion à son cluster, il les confirme et il attribue un slot et un code CDMA pour chaque nœud accepté pour pouvoir envoyer ses données dans son slot et avec son code pour éviter les collisions.
- Après la réception du slot et du code CSMA par chaque nœud demandeur d'adhésion à un cluster, les clusters seront formés et la phase d'initialisation sera achevée.

b) La phase transmission

- chaque nœud doit attendre son tour, selon les slots, pour envoyer ses données (la donnée captée et son énergie résiduelle).
- Les Clusters Head reçoivent les données captées et l'énergie résiduelle des nœuds, agrègent la donnée selon une fonction d'agrégation (moyenne, somme, suppression des redondances…etc.) et gardent les informations (ID, énergie, slot, code) du nœud ayant une énergie résiduelle plus grande pour le désigner comme le prochain Cluster Head.
- Le nœud puits reçoit les données agrégées ainsi que les ID des prochains Clusters Head envoyées par les nœuds-chef.
- A la fin de la durée du la phase de transmission, le nœud puits lance le déclenchement du nouveau round et informe les nœuds sur l'identité de leurs prochains Cluster Head selon la dernière mise à jour (dernière agrégation reçue) effectuée.

2- round ≠ 0

C'est le cas des rounds qui suivent le premier round, et comme il n'existe pas une phase d'initialisation, puisque la structure du réseau ne change pas, il n'y a que la phase de transmission.

- Chaque nœud reçoit le déclanchement du nouveau round, vérifie s'il est choisi comme Cluster Head pour ce round, sinon change l'adresse de Cluster Head.
- Après que tous les nœuds exécutent le déclenchement du nouveau round, chaque nœud commence à envoyer ses données selon son slot et son code attribué.

5. Vue globale de la solution proposée

Dans cette section, nous présenterons une vue globale de la solution que nous avons proposé afin de sécuriser le protocole MH-LEACH PSM. On va donc intégrer à ce dernier des mécanismes simples et robustes pour la détection des nœuds attaquants. Dans notre étude, nous nous intéressons aux attaques de type trou noir et trou de base.

Dans le premier type d'attaque, qui est trou noir, le nœud attaquant se positionne comme CH dans le réseau, il reçoit ainsi toutes les données collectées par les nœuds capteurs de son cluster. Ensuite, au lieu de les transmettre vers la station de base, il supprime ces données reçues et ils les empêchent ainsi d'atteindre la station de base. Par conséquent, ces nœuds attaquants peuvent créer un disfonctionnement total du RCSF hiérarchique.

Après avoir étudié le comportement de ce type d'attaques, la solution la plus appropriée est de surveiller tous les CH. En partant de l'hypothèse que tous les nœuds membres captent des données, au bout d'un round, si la station de base ne reçoit aucun résultat d'agrégation d'un CH donné, elle le considère comme nœud suspect et elle le met en quarantaine.

Pour le deuxième type qui est trou de base, le nœud attaquant se positionne comme station de base, avec un signale plus grand que celui de la station de base d'origine, donc il attire tout le trafic vers lui.

La solution la plus adéquate pour ce type d'attaque est la vérification de l'authentification des nœuds, en comparant le code mac de l'émetteur à celui de la station de base, si les deux valeurs sont différentes, alors le nœud sera ajouté à la liste des nœuds suspects.

6. Attaque Trou Noir

6.1. Implémentation de l'attaque Trou Noir

Pour implémenter cette attaque, nous partons sur l'hypothèse qu'un nœud attaquant dispose d'une ressource énergétique élevée, et d'une puissance de signal supérieure à celle d'un nœud légitime. Ce niveau d'énergie lui permettrait d'avoir une durée de vie maximale dans l'objectif de perturber le réseau de capteurs le plus longtemps possible. La puissance du signal lui permettrait de former des clusters avec le maximum de nœuds et avoir ainsi accès à un maximum de données. L'algorithme qui va suivre décrit le comportement d'un nœud attaquant par rapport à un nœud légitime.

Notations utilisé :

Notation	Description
CHi	Cluster Head
BS	Base Station
N	Nombre total de nœuds
CM	Membre d'un cluster
Ni	Nœud i
X (n i)	Nombre aléatoire généré par ni
T (ni)	Seuil calculé par chaque nœud.

Mal	Nœud malicieux
Pro_ID_ch	Identificateur du prochain CH
Black_list	Liste des nœuds suspects
Size	Taille de la liste noire
Msg_env	Nombre de message envoyé par un CH
mac	Le code mac calculé par un nœud
R	Le round courant

- **Algorithme de modélisation de l'attaque trou noir**

Phase désignation CH :

If ni== mal **then**

CH = ni

else

∀i ϵN-{Mal} calcul X (n i) et T (ni)

If X (ni) < T (ni) **then**

CHi = ni

endif

endif

CH diffusent les Messages ADV

Tous les CM vont joindre leurs CH

Phase agrégation:

CH crée la table TDMA

CM envoient les données au CH

If CHi == mal **then**

Supprime tous les messages reçus

else

Agrégation des données et routage vers la BS

endif

6.2. Détection de l'attaque Trou Noir

La solution la plus intuitive pour la détection de l'attaque trou noir est de mettre en place une surveillance du comportement des nœuds CH et cela en se basant sur le comptage par la station de base du nombre de messages reçus de chaque CH .Dans le premier round, la station de base fixe un compteur pour chaque CH ,ce dernier est incrémenté à chaque fois que la station de base reçoit un message du nœud CH associé à ce compteur. A la fin du round la station de base vérifie tous les compteurs et si l'un d'eux est égal à zéro, donc anomalie dans le réseau c'est-à-dire que le nœud CH à supprimer toutes les données reçues par les nœuds de son cluster. Ce dernier est considéré comme trou noir. Après la détection de cette attaque le nœud malicieux sera ajouté à une liste noire conçue pour les nœuds qui représente une menace pour le réseau. De cette manière, on a réussi à isoler le trou noir mais pour être sûr que le nœud malicieux ne sera pas élu CH dans les prochains rounds, on fait un test. Pour se faire avant le déclenchement du nouveau round, on parcoure la liste noire et si le prochain nœud choisi par la station fait partie de cette liste, il sera remplacé directement par un autre nœud aléatoirement.

- **Algorithme de détection du trou noir**

```
//fin du round//
//La station de base vérifie le nombre de messages reçu par chaque nœud CH//
if (CHi) et (msg_env=0) then
//le nœud CHi est suspect, il sera ajouté à la liste noir//
black_list[]=CHi;
endif

Mise en quarantaine des nœuds suspects
 z=0;
Trouve=FALSE;
while ((z<size)&&(trouve==FALSE))
Begin
if (Pro_ID_ch ϵ black_list[]) then
Trouve=TRUE;
Le prochain CH est suspect
Pro_ID_ch= Random.rand()%N;
Il sera remplacé par un nœud choisi aléatoirement
endif
z++
end
```

7. Attaque Trou de Base

7.1. Mise en œuvre de Trou de Base

Dans notre cas, on a implémenté cette attaque de telle façon que le nœud malicieux se fait passer pour une station de base en envoyant un signal plus fort que celui de la station de base d'origine, il désigne une valeur de probabilité très élevé pour augmenter le nombre des CH, ceci provoque beaucoup de messages échangés correspondants aux invitations envoyées par les CH aux nœuds membres pour rejoindre leur clusters, les demandes d'admission, l'envoi des slots et du code CDMA etc…

7.2. Détection de Trou de base

La solution que nous avons proposée pour détecter les trous noirs, répond parfaitement à nos besoins et à la problématique que nous avons fixée au départ. Pour rendre le protocole plus efficace et assurer tout de même sa stabilité et sa continuité, surtout devant les attaques externes au réseau, on a implémenté un mécanisme simple basé sur l'authentification et l'intégrité des données afin d'empêcher un attaquant d'emprunter l'identité des nœuds légitimes pour s'approprier leurs données. Pour cela on a utilisé des outils cryptographiques afin de sécuriser la communication dans le réseau et bloquer ainsi toute tentative de perturbation du réseau.

Ce nouveau mécanisme implémenté permet d'assurer la sécurité des communications dans les différents liens (Puits-Membre, Membre-CH, CH-Puits…etc.) d'une manière similaire, c'est-à-dire le mécanisme fonctionne de la même sorte pour tous les liens du réseau.

La solution est présentée sous forme de code d'authentification de message MAC qui sera calculé par une fonction de hachage mais sans l'utilisation de clés symétriques qu'on a préféré éviter, malgré leur efficacité et leur fiabilité en terme de sécurisation des liens, mais qui conduisent à une gestion des clés, différente pour chaque lien, qui nécessite plus d'espace mémoire et peut, par conséquent, augmenter la complexité du programme avec une dissipation d'énergie additionnelle suite aux échanges des tables des clés entre les différents liens.

Le calcul du message d'authentification MAC dépend de l'ID de l'émetteur et le round courant (lors la transmission et/ou la réception) d'où chaque nœud voulant émettre un message doit calculer ce message en utilisant cette fonction :

$$Mac\ (m) = (IDemt+1)*(round\_courant+1)\ mode\ N$$ où :

N : nombre totale des nœuds capteurs du réseau.

Le récepteur recalcule le MAC en utilisant le contenu du message reçu (IDemt), et si le mac recalculé est égal au mac reçu dans le message, le nœud accepte ce message et sera traité, sinon il le rejette et considère l'émetteur comme un attaquant.

- **Algorithme de détection du Trou de Base**

Envoi d'un paquet

//Calcul du MAC sur le paquet de l'émetteur //

mac=((ID_Emetteur)+1)*(r+1))%N

//Envoi du paquet //

DATA->mac=Mac,
DATA->ID=ID_Emmeteur,
DATA->Round=R,
Send(DATA)

Réception du paquet

mac=DATA->mac

ID_Emetteur=DATA->ID

R=DATA->Round

//Calcul du MAC en utilisant le contenu du message reçu//

MAC=((ID_Emetteur)+1)*(r+1))%N

//Vérifier si le nœud émetteur est authentique//

if MAC==mac **then**

//L'authentification est correcte//

else

//L'authentification est incorrecte//

//ajout du nœud émetteur à la liste noir

black_list[]=ID_Emetteur

endif

8. Conclusion

Etant un protocole de routage non sécurisé, MH-LEACH PSM est vulnérable à de dangereuses attaques. Dans ce chapitre, nous avons présenté les différents mécanismes qui permettent à MH-LEACH PSM de faire face à l'attaque Trou Noir et Trou de Base. Cela nous a permis de concevoir un protocole sécurisé. Pour cela 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.

Nous allons passer à l'implémentation de toutes les étapes de notre solution et donner des résultats démonstratifs qui justifient son efficacité. Le prochain chapitre sera consacré à la mise en œuvre de notre solution.

1 Introduction

Tel qu'on l'a montré au cours du chapitre précédant, l'objectif principal de notre travail est la réalisation d'une solution qui se charge de sécuriser le protocole de routage MH-LEACH PSM. Notre premier but est d'atteindre un niveau de sécurité acceptable sans dégrader les performances du réseau. Sachant que la sécurité et la consommation d'énergie sont deux facteurs très importants qu'il faut prendre en considération lors du déploiement d'un RCSF, et ainsi faire le maximum d'efforts afin de sécuriser les communications pour assurer une consommation réduite de l'énergie pour le bon fonctionnement du réseau entier. De ce fait, nous avons établi un nouveau protocole qui est en mesure de pallier à l'une des importantes attaques visant le protocole de routage LEACH .L'objectif de ce chapitre est donc de démontrer l'efficacité du protocole proposé par rapport au protocole MH-LEACH PSM en termes de sécurité ainsi que d'autres métriques de performances via l'implémentation et la simulation des deux protocoles. Pour cela, nous commencerons par définir les outils nécessaires pour l'implémentation et la simulation des deux protocoles. Ensuite, nous décrirons la mise en œuvre de tous les processus décrits lors de conception de la solution qui est le chapitre 3. Nous terminerons par une présentation des résultats relevés lors des tests de performances sur MH-LEACH PSM et notre protocole sécurisé.

2 Environnements de simulation

Les RCSF nécessitent une phase de test avant la mise en place. Pour cela, la solution la plus fiable et la moins coûteuse consiste en « la simulation ». La simulation des RCSF consiste principalement à la reproduction du comportement des nœuds capteurs du monde réel dans le monde virtuel. Cette étape de test avant la mise en place du réseau a pour objectif de tester l'efficacité réelle des protocoles de routage et de sécurité développés et de connaitre au préalable leurs failles et limitations. Une condition nécessaire au bon fonctionnement des tests serait d'exécuter les instructions du code sur une machine disposant d'un environnement d'exécution similaire à celui du capteur. Cette condition est garantie par un système d'exploitation spécialement dédié aux RCSF, c'est-à-dire un ensemble de fonctions, de procédures et de mécanismes produisant un code minimisé respectant les ressources limitées en mémoire et en capacités de calculs des capteurs. Plusieurs systèmes ont été proposés pour les réseaux de capteurs sans fil : TinyOs, SOS **[89]**, Contiki **[88]**, MANTIS**[90]**.…etc. TinyOs est le plus populaire et le plus utilisé, pourquoi nous l'avons choisi pour nos ateliers pratiques dans la simulation ?

2.1 TinyOs

TinyOs [83] a été développé par l'université de Berckley. Principalement dédié aux systèmes embarqués, il s'agit d'un système open source complètement écrit en langage C et ses composants ont été ré-implémentés en langage Nesc. Il se distingue par [85]:

- Une taille de mémoire réduite.
- Une basse consommation d'énergie.
- Des opérations robustes.
- Applications orientées composants: TinyOS fournit une réserve de composants systèmes utilisables au besoin.

- Programmation simple et puissante orienté évènement : Généralement sur TinyOS, un programme s'exécute suivant le déclenchement des événements. Sinon, les capteurs restent en veille ce qui maximise la durée de vie du réseau.
- Un code portable sur les différentes plateformes existantes.

2.1.1 Notions principales

TinyOS est construit autour des différents concepts décrits ci-dessous: **[83]**

- Les composants : constitués de :
- Frame : est un espace mémoire de taille fixe permettant au composant de stocker les
- variables globales et les données qu'il utilise. Il n'en existe qu'un seul par composant.
- Tâches : contiennent l'implémentation des fonctions. Elles sont décomposées en deux

catégories: les commandes et les évènements.

- Les interfaces **:** représentent le descriptif des fonctions définies dans les tâches.

2.1.2 Langage NesC

NesC est un langage de programmation orienté composants syntaxiquement proche du langage C. Il est conçu pour la réalisation des systèmes embarqués distribués, en particulier, les RCSF. **[91]**Il existe trois types de fichiers sources des applications NesC : les fichiers interfaces et les fichiers configurations et modules qui constituent les composants. **[93].**Les modules définissent le code de l'application et implémentent des interfaces qui représentent l'unique point d'accès au composant. Les configurations permettent d'assembler les composants en reliant les interfaces du composant aux interfaces des autres composants [4]. Une interface définit d'une manière abstraite les interactions entre deux composants. Elle définit un fichier décrivant les commandes et les évènements proposés par le composant qui les implémente. Une commande doit être implémentée par le fournisseur de l'interface et un évènement doit être implémenté par l'utilisateur de l'interface. [51]

L'avantage de programmer en Nesc est la possibilité de construire des composants pouvant être des systèmes complets.

2.2 Les simulateurs

Dans ce qui suit on va voir les différents simulateurs développé par TinyOs :

2.2.1 TOSSIM

Pour arriver à simuler le comportement des capteurs au sein d'un RCSF, un outil très puissant a été développé et proposé pour TinyOS sous le nom de TOSSIM. Le principal but de TOSSIM est de créer une simulation très proche de ce qui se passe dans les RCSF dans le monde réel. Une économie d'effort et une préservation du matériel sont possibles grâce à cet outil. **[84]**

Pour une compréhension moins complexe de l'activité du réseau, TOSSIM peut être utilisé avec une interface graphique TinyViz. Cette dernière est équipée par plusieurs API plugins qui permettent d'ajouter plusieurs fonctions à notre simulateur comme par exemple suivre la dépense d'énergie en utilisant un autre simulateur qui s'appelle PowerTOSSIM. **[85]**

2.2.2 TinyViz

TinyViz [86] est une interface graphique Java. Elle permet de donner un aperçu des capteurs à tout instant ainsi que des divers messages qu'ils émettent. Elle détermine un délai entre chaque itération des capteurs afin de permettre une analyse pas à pas du bon déroulement des actions en activant différents modes comme Radio, CPU, etc. [86]

2.2.3 PowerTOSSIM

Le simulateur TOSSIM n'a pas la capacité de vérifier le taux d'énergie dissipée pendant l'exécution des applications. Cependant, le besoin de vérifier la consommation énergétique dans un RCSF a un intérêt primordial. L'université de Harvard a conçu le simulateur PowerTOSSIM qui surmonte ce problème. Ce nouveau simulateur est intégré dans TOSSIM. Il permet de générer un fichier de l'extension **.**trace qui enregistre les détails de la simulation comme l'énergie consommée dans le réseau.

3 Implémentations et déroulements

Dans cette partie, nous expliquons et déroulons les phases de l'algorithme LEACH et de sa variante qui est MH-LEACH PSM. On fait appel à l'interface graphique TinyViz pour visualiser le déroulement de la simulation, un fichier de configuration est créé qui permet à TinyViz de se lancer avec des paramètres spécifiés. Ces derniers représentent : le nombre et l'emplacement des capteurs, la durée de la simulation et les plugins que nous souhaitons activer dès le début de la simulation comme Debug Messages.

Afin de sécuriser le protocole LEACH, notre travail s'est déroulé en quatre phases :

1) Implémentation du protocole LEACH.
2) Implémentation du protocole MH-LEACH PSM.
3) Implémentation de la solution proposée.

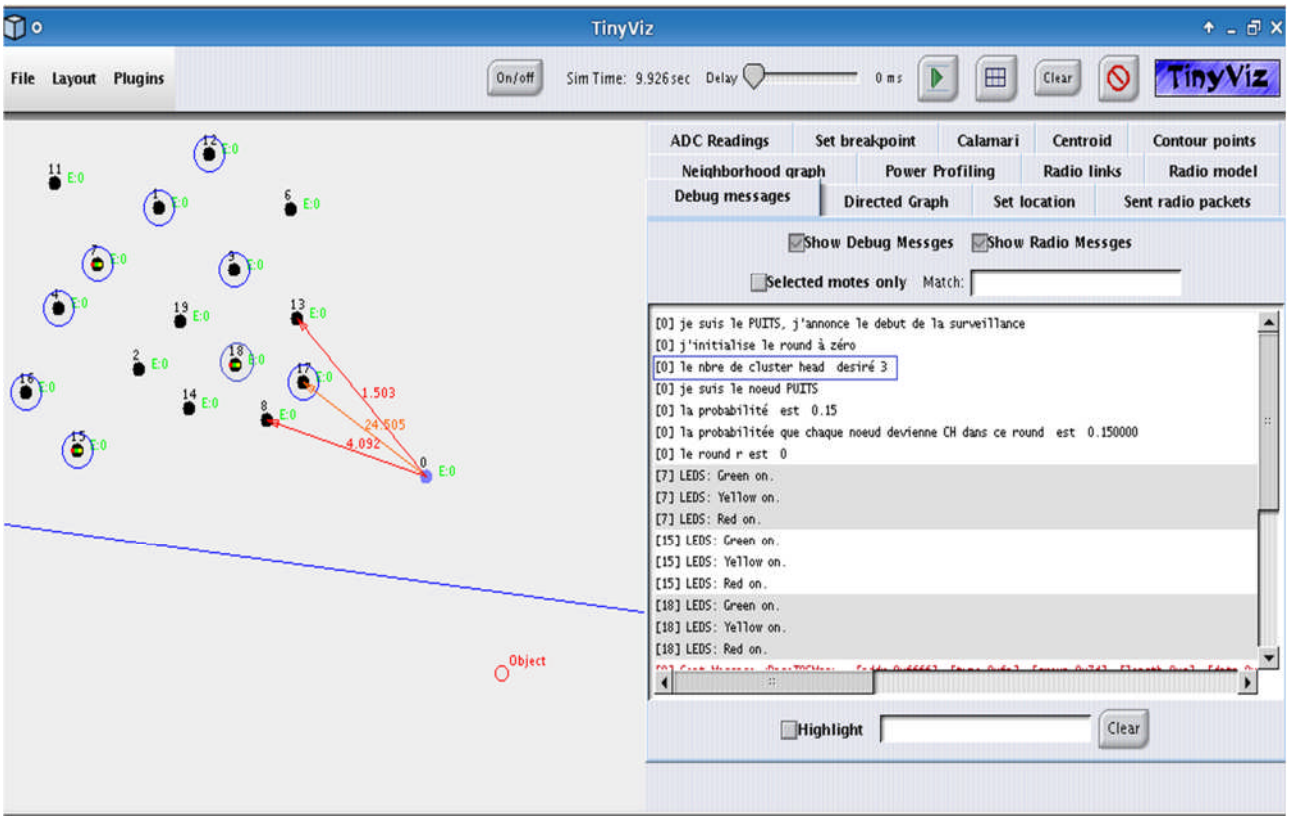
Nous donnons donc un aperçu de notre implémentation afin de voir l'avantage de MH-LEACH PSM par rapport à LEACH et son évolution après la mise en place de quelques mesures de sécurité contre un certain type d'attaque qu'on va voir par la suite.

3.1 Déroulement du protocole LEACH :

Nous présentons dans ce qui suit l'implémentation du protocole de routage LEACH :

3.1.1 Déclenchement et relai du nouveau round, et, annonce des CH:

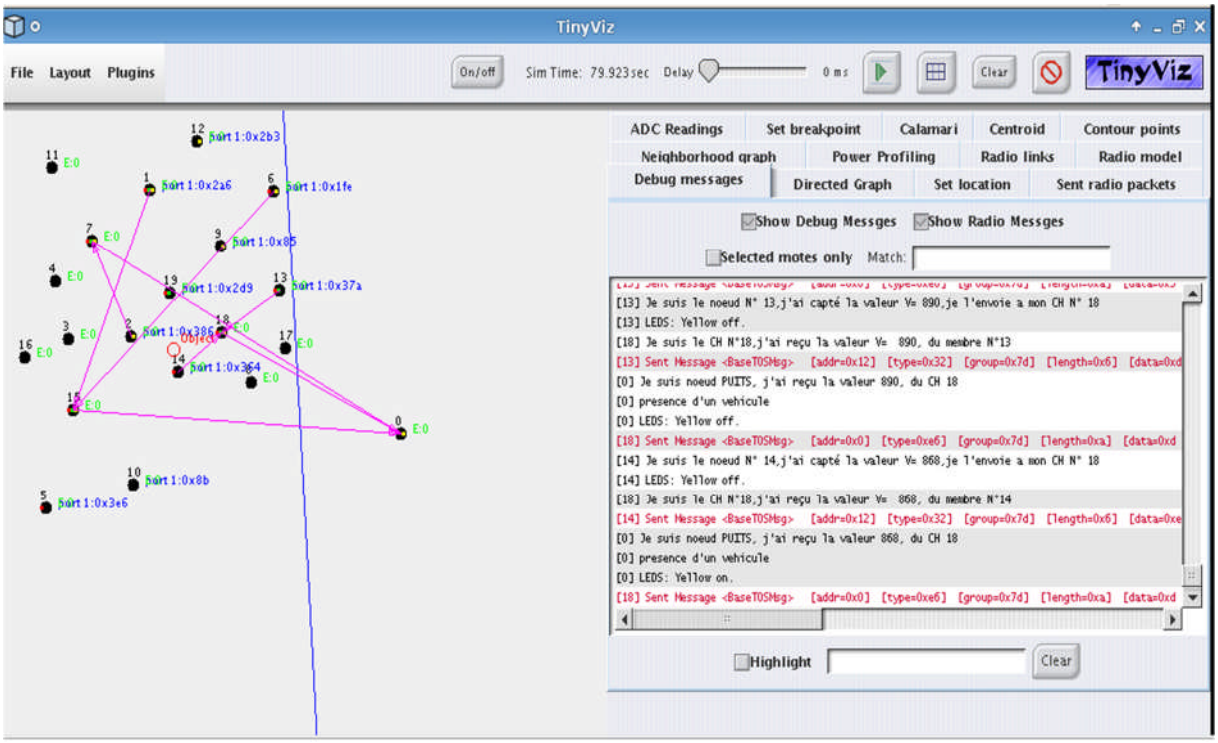
La figure IV.1 représente les transmissions par diffusion qui se passent durant différentes étapes de l'algorithme LEACH. Une transmission par diffusion est repérée par un cercle bleu. La station de base envoie un message par diffusion aux nœuds voisins pour l'annonce du round et le nombre de CH désiré. Ces voisins prennent le relai en envoyant à leur tour. De plus, nous pouvons voir que les nœuds 7,15et 18 seront élus CH. Cet évènement est marqué par l'activation des LED rouges des CH. Ensuite, le CH 7, CH 15 et CH 18 diffusent des annonces pour signaler leur statut.



FigureIV.1 : Déclenchement et relai du nouveau round, annonce des CH 7, CH 15, CH18.

3.1.2 Formation de groupes et envoi des températures au nœud puits :

La figure IV.2 représente quelques transmissions unicast qui se passent durant différentes étapes de l'algorithme LEACH. Une transmission unicast est repérée par une flèche. Durant la première étape, les nœuds non-CH répondent à l'annonce des CH les plus proches. La figure IV.2 illustre la formation du CH 7, CH 15 et CH 18**.** Quant à la seconde étape, chaque membre capte la température et attend le début de son slot pour qu'il puisse l'envoyer à son CH. La troisième étape représente l'envoi des résultats d'agrégation des températures reçues par chaque CH au nœud puits.



FigureIV.2 : Formation de groupes et envoi des résultats d'agrégation à la station de base.

3.2 Déroulement du protocole MH-LEACH PSM :

Le protocole MH-LEACH PSM comme on l'a vu dans le chapitre précédant est une variante améliorée de LEACH .Pour voir la principale différence entre LEACH et sa variante on va dérouler dans ce qui suit la variante MH-LEACH PSM.

3.2.1 Déclenchement du round et annonce des CH :

La station de base initialise le round à 0 et choisi aléatoirement les CH selon le nombre des clusters désiré. La figure IV.3 illustre le nœud14 comme CH du cluster 1, la station de base lance un message de déclenchement du round 0 aux autres nœuds du réseau.

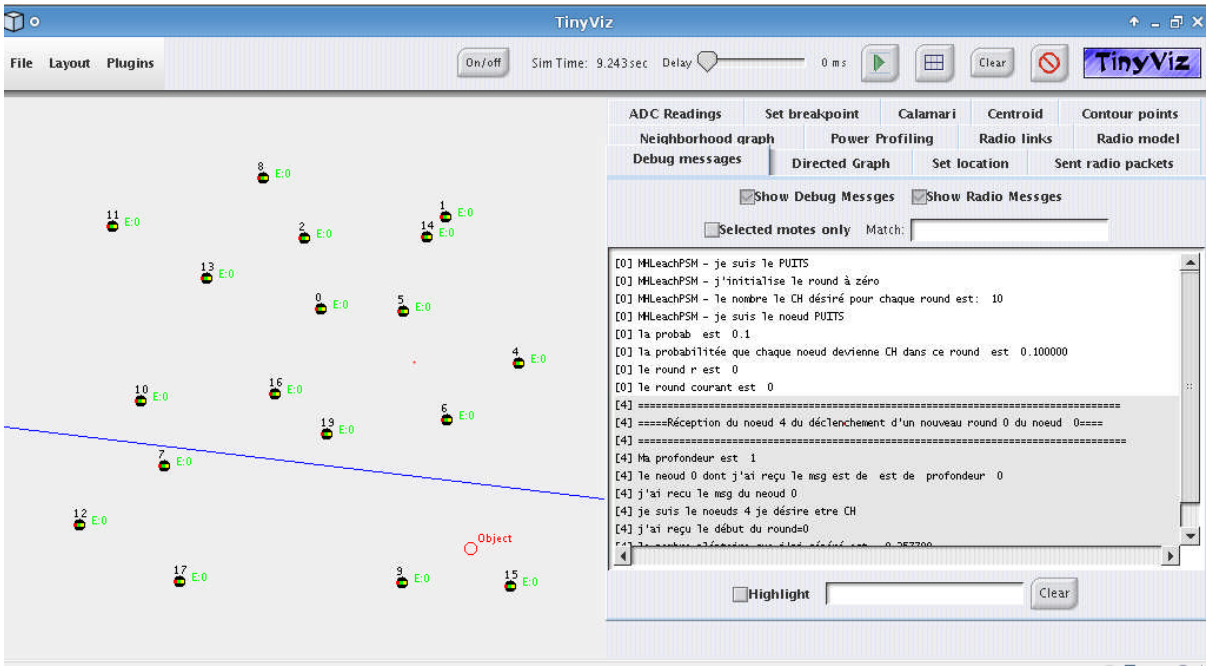


Figure IV.3 : Déclenchement du round et annonce des CH choisi.

3.2.2 Réception des données par le CH

On voit sur la figure IV.4 les CH 7 et 15 reçoivent les données captées et l'énergie résiduelle des nœuds, ils agrègent les données selon une fonction d'agrégation et envoi le résultat à la station de base .Le nœud ayant une énergie résiduelle plus grande sera désigner comme le prochain Cluster Head et dans ce cas c'est le nœud 11.

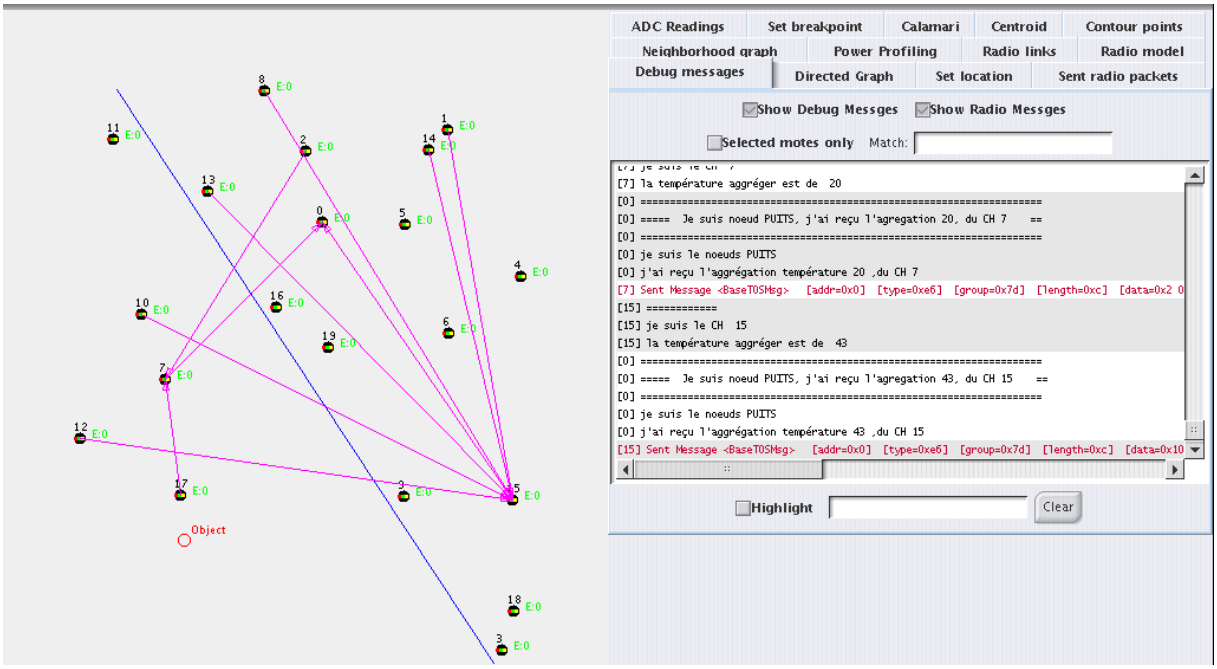


Figure IV.4: Agrégation et envoi des résultats à la station de base.

4 Implémentation de attaque Trou Noir sur notre protocole

Dans cette partie on va tester la réaction de notre protocole contre l'attaque Trou Noir, une attaque dangereuse visant le protocole MH-LEACH PSM. Dans ce qui suit on va d'abord vérifier ses effets sur le déroulement de notre protocole. Une fois l'attaque détectée on va y remédier grâce à notre solution.

4.1 Déclenchement du round 0 et annonce des CH

On remarque sur la figure IV.1 le déroulement normal de notre protocole, la station de base annonce le CH 5 pour le cluster1 et CH10 pour le cluster 2.

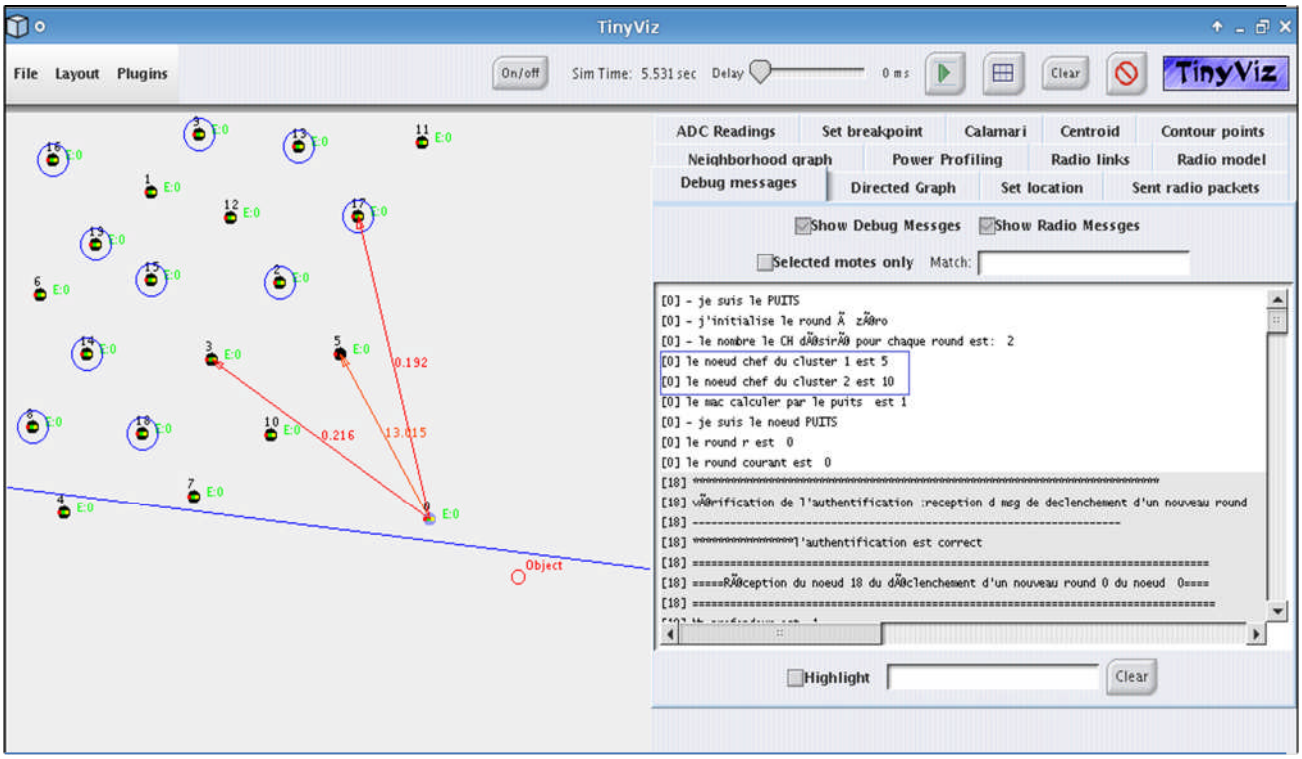


Figure IV.5 : Déclenchement du round 0 et annonce des CH 5 et CH 10

4.2 Détection de l'attaque trou noir

La Figure IV.2 montre l'envoi du résultat d'agrégation des données du CH 10 au nœud puits mais dans le cas de CH 5 aucun message envoyé à la station de base donc anomalie dans le réseau, le nœud 5 à supprimer toutes les données reçues par les nœuds de son cluster donc il est considéré comme trou noir.

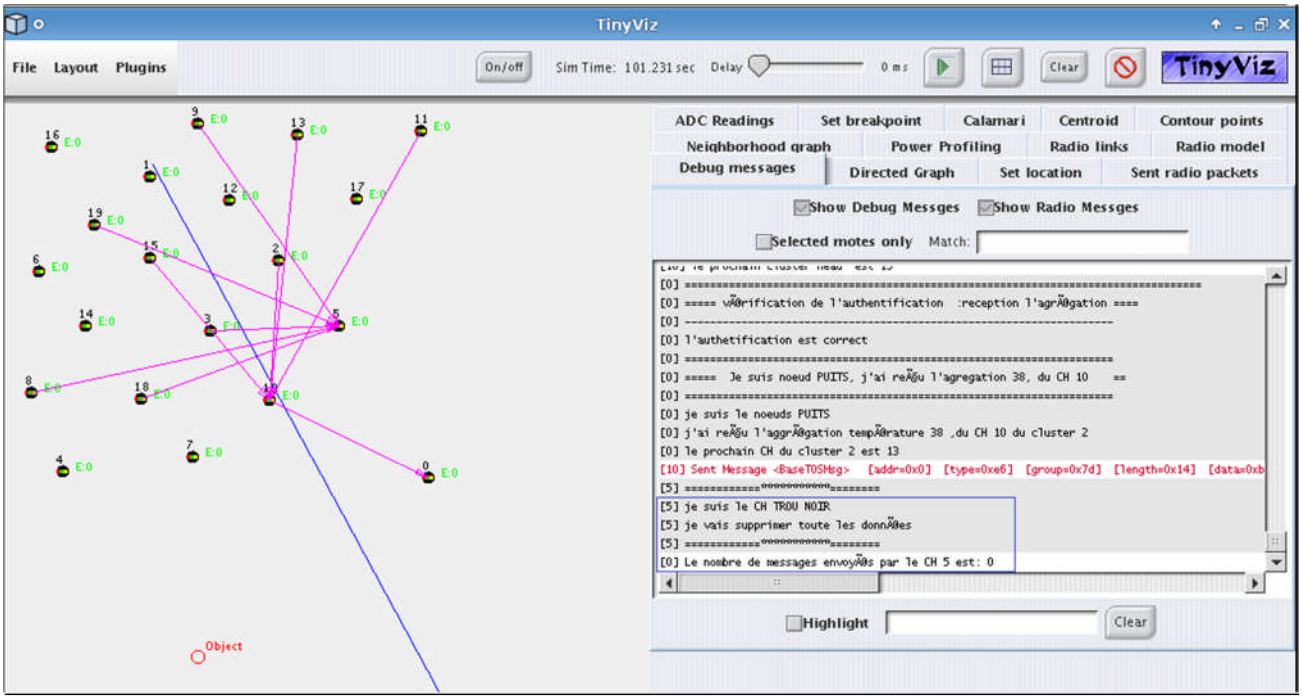


Figure IV.6 : Détection du trou noir.

4.3 Mise en quarantaine du nœud suspect et son remplacement

On a réussi à détecter le nœud suspect qui est le CH 5ce qui veut dire que le nombre de messages envoyé à la station de base est zéro donc le nœud 5 sera mis en quarantaine. Et sera remplacé par le nœud 3 dans le prochain round comme l'illustre la figure IV.7.

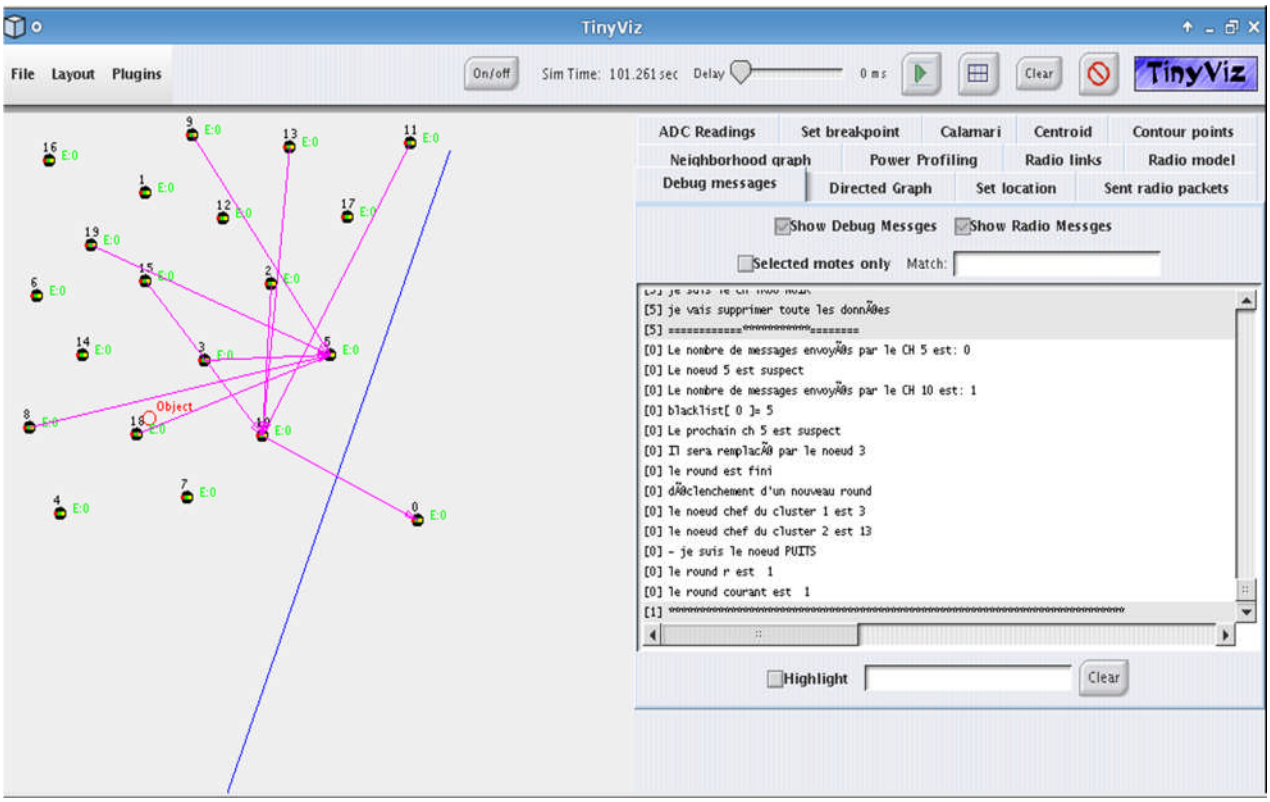


Figure IV.7 : Mise en quarantaine du trou noir et son remplacement.

4.4 Vérification de la solution

Dans cette figure on remarque que le CH 3 choisi pour remplacer le trou noir mis en quarantaine, marche correctement d'où l'efficacité de notre solution et la robustesse de notre protocole.

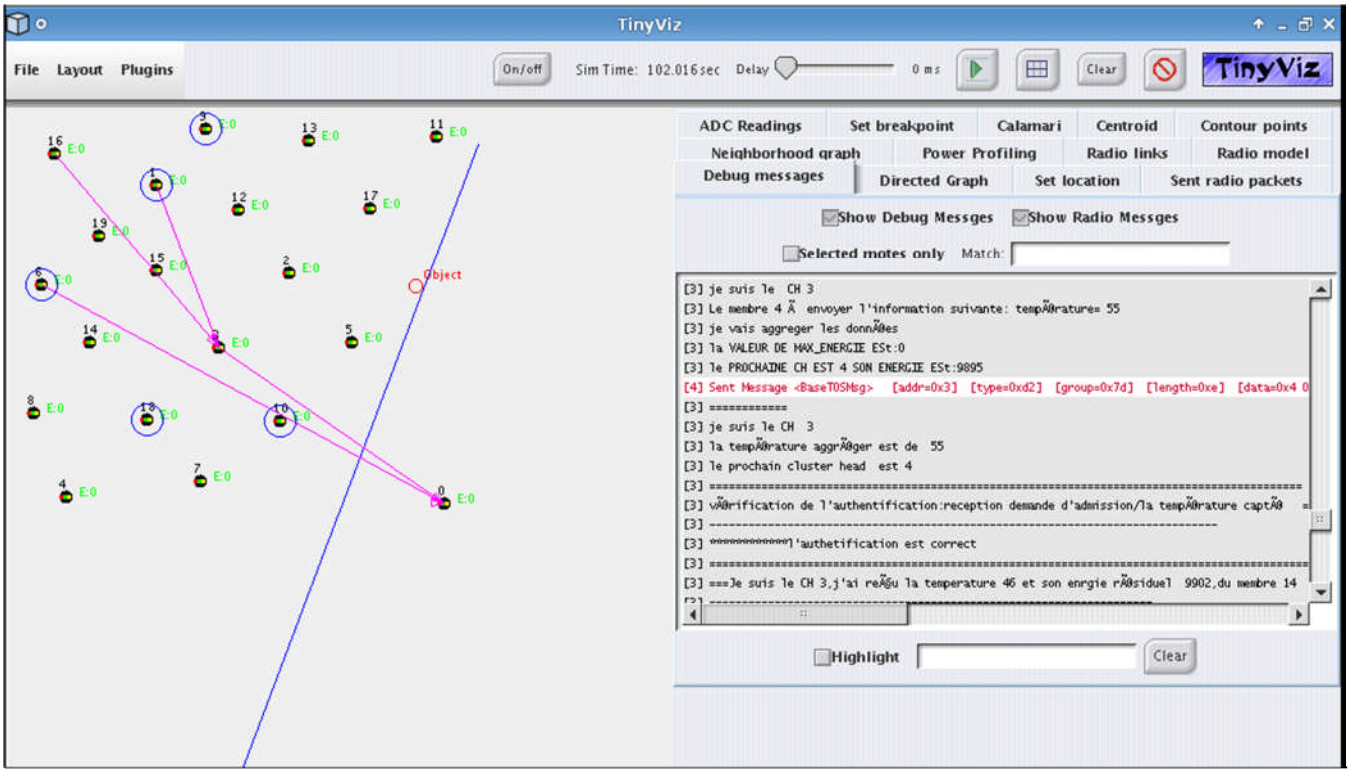


Figure IV .8 : Envoi des informations du CH 3à la station de base.

5. Implémentation de l'attaque de Trou de Base sur notre protocole

Dans cette partie on va tester la réaction de notre protocole contre l'attaque Trou de Base, une attaque dangereuse visant le protocole MH-LEACH PSM. Dans ce qui suit, on va d'abord vérifier ses effets sur le déroulement de notre protocole. Une fois l'attaque détectée on va y remédier grâce à notre solution.

5.1.Déclenchement du round 0 et annonce des CH

On remarque sur la figure IV.9 le déroulement normal de notre protocole, le nœud puits annonce le début du round.

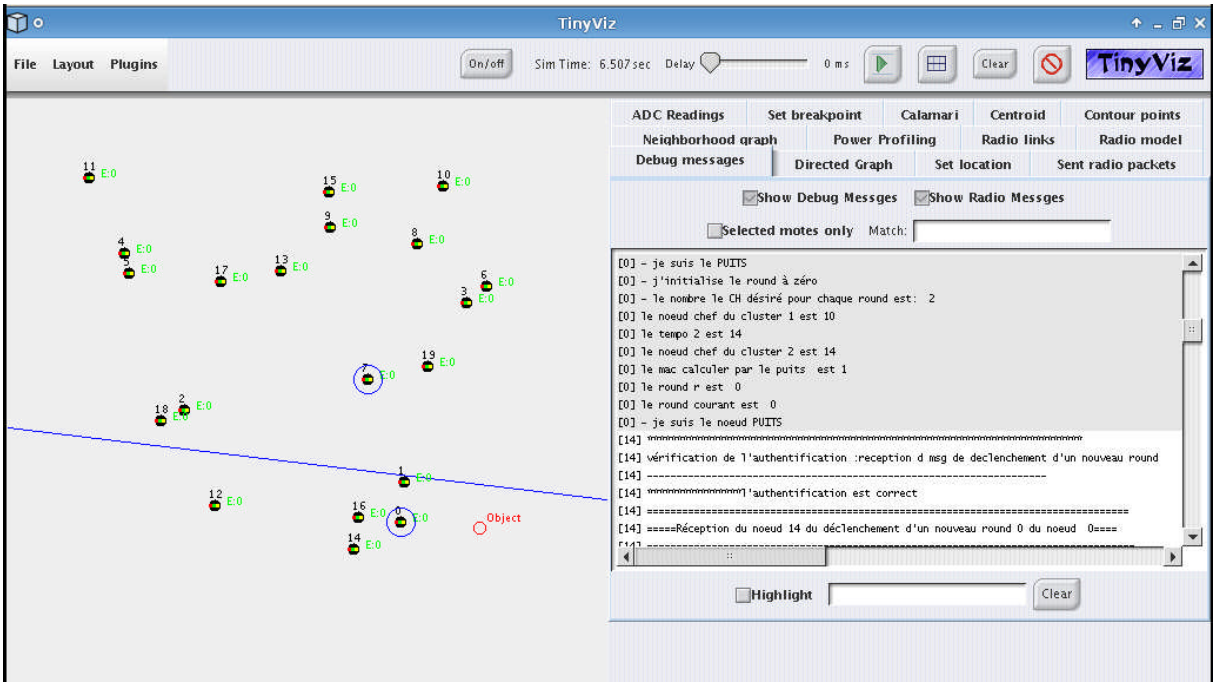


Figure IV.9 : Déclenchement du round

5.2. Détection de l'attaque Trou de Base

On remarque sur la figure IV.10, le nœud attaquant, dès qu'il annonce le début du round, il est directement détecté par les nœuds 19 et 18.

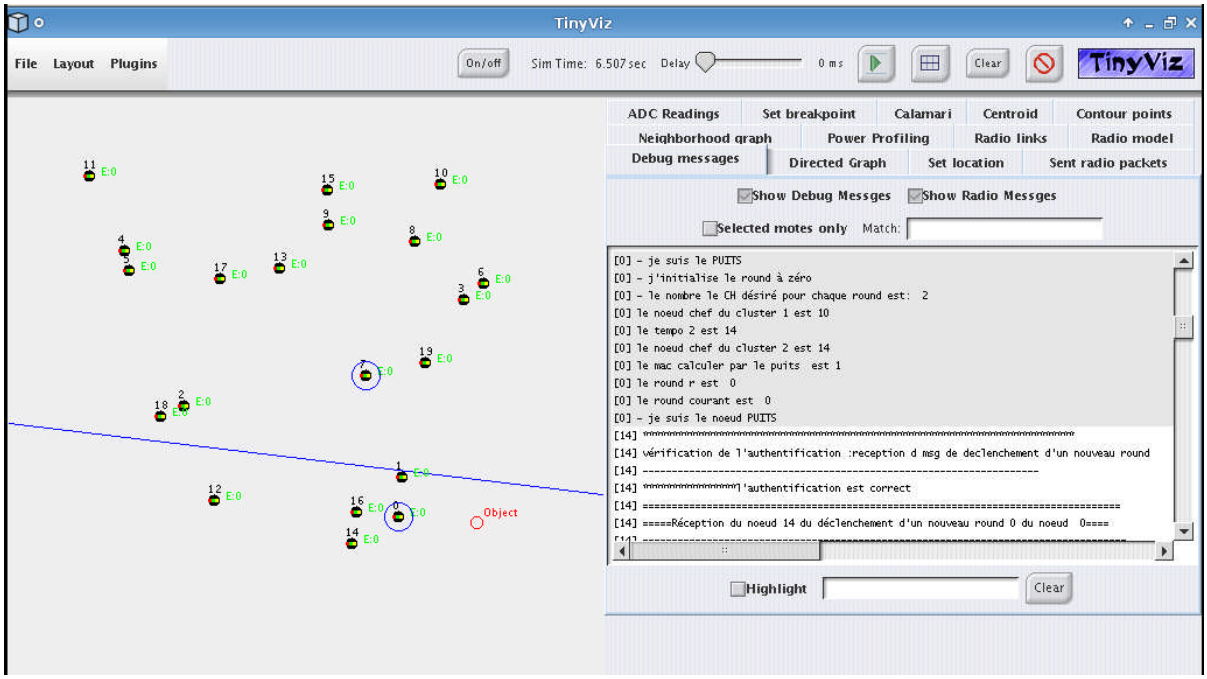


Figure IV.10 : Détection de l'attaque Trou de base.

5.3. Mise en quarantaine du nœud suspect

On remarque sur la figure IV.11 que le nœud 7, qui est le nœud attaquant, sera ajouté à la liste des nœuds suspects.

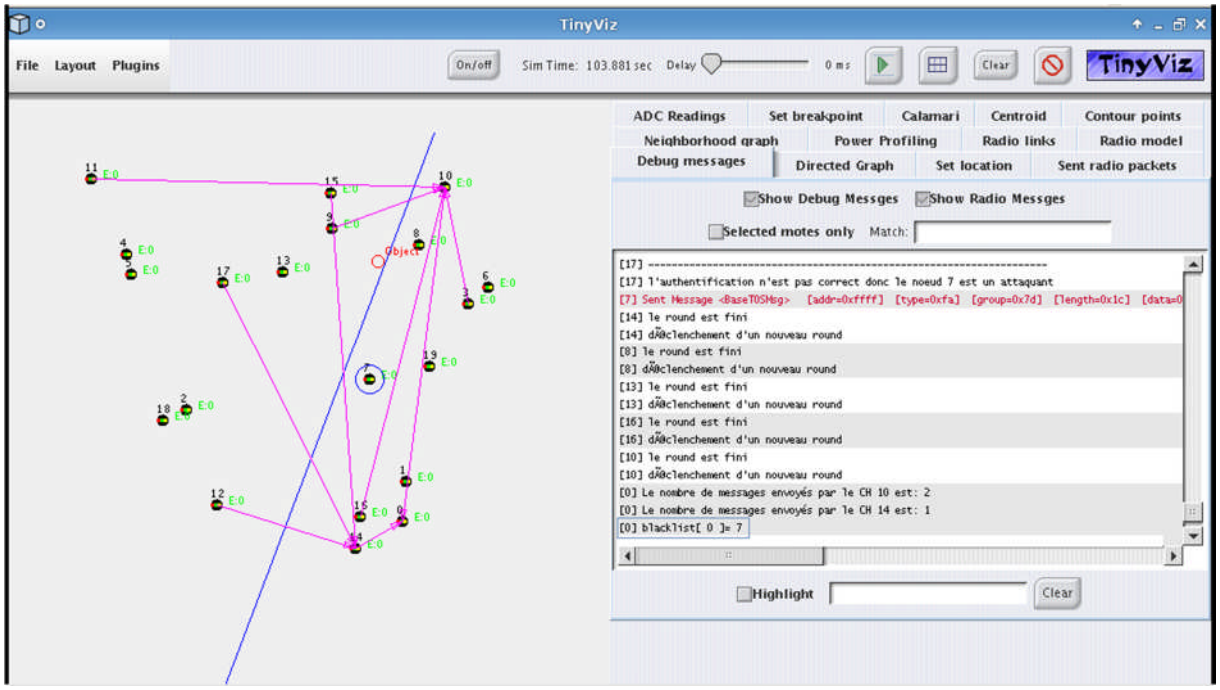


Figure IV.11 : Mise en quarantaine du nœud Trou de base.

6. Résultats et Performances

Pour évaluer les performances de notre protocole sécurisé, nous avons procédé à le comparer au protocole de routage MH-LEACH PSM. Pour cela, nous avons effectué des simulations avec les mêmes paramètres et métriques pour les deux protocoles. De plus, nous avons étudié l'effet des attaques trou noir e trou de base sur les performances des deux protocoles afin de vérifier l'efficacité de notre protocole.

6.1 Métriques à évaluer

Pour pouvoir comparer les performances de notre protocole avec celle de MH-LEACH PSM, il est commode de mesurer certaines métriques qui sont :

6.1.1 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. Nous analysons donc l'impact de mécanismes de sécurité intégrés dans notre protocole sur l'énergie consommée par rapport au protocole MH-LEACH PSM.

Pour se faire, nous prenons comme critère, l'énergie moyenne consommée par chaque nœud du réseau. De plus, nous vérifions, pour les deux protocoles, l'effet dégradant de la consommation d'énergie due à l'attaque Trou noir.

6.1.2 Perte de paquets

Les protocoles de routage utilisent cette métrique dans le but de minimiser le nombre de paquets de données perdus lors du transfert depuis une source vers une destination pendant le routage [**95]**.
Le choix de cette métrique, comme étant un critère de performance, revient à sa nécessité dans certaines applications où les données échangées sont très critiques. Pour la mesurer, nous

calculons la moyenne des taux de perte de paquets de températures entre les membres et leurs CH, et de paquets d'agrégation de ces températures entre les CH et la station de base. Ainsi, notre protocole ne doit pas mener à une forte perte de paquets de données par rapport à MH-LEACH PSM. De plus, nous vérifions, pour les deux protocoles, l'effet de l'attaque Trou noir sur l'augmentation de nombre de paquets de données perdus.

6.1.3 Délai de bout-en-bout

L'EED (End-to-End Delay) est le temps moyen nécessaire pour qu'un paquet de données soit acheminé à partir de la source vers la destination [95]. Cette technique est parmi les métriques les plus connues dans les réseaux sans fil. Les protocoles de routage l'utilisent pour minimiser le temps de propagation des paquets de données échangés pendant le routage. Le critère que nous utilisons est l'EED moyen de tous les paquets transitant dans le réseau.

6.2 Paramétrage de la simulation

Avant de lancer les simulations, nous devons ajuster certains paramètres qui sont présentés par le tableau IV.2 :

Paramètres du contexte de la simulation	
Nombre de nœuds du réseau	20
Délai de la simulation	7 minutes
Placement des nœuds	Aléatoire
Nombre de stations de bases	1
Durée d'un round	7 minutes
Nombre d'itérations	5 : les résultats que nous allons présenter sont une moyenne de 5 simulations pour un même scénario
Taille de paquet de données	29 octets : c'est le paquet de transmission de TinyOS
Modèle de propagation	Modèle Lossy

Tableau IV. 2: Paramètres du contexte de la simulation.

6.3 Résultats et interprétations

Dans ce qui suit, nous allons présenter et analyser les résultats de simulation obtenus, suivant les métriques de performances citées précédemment.

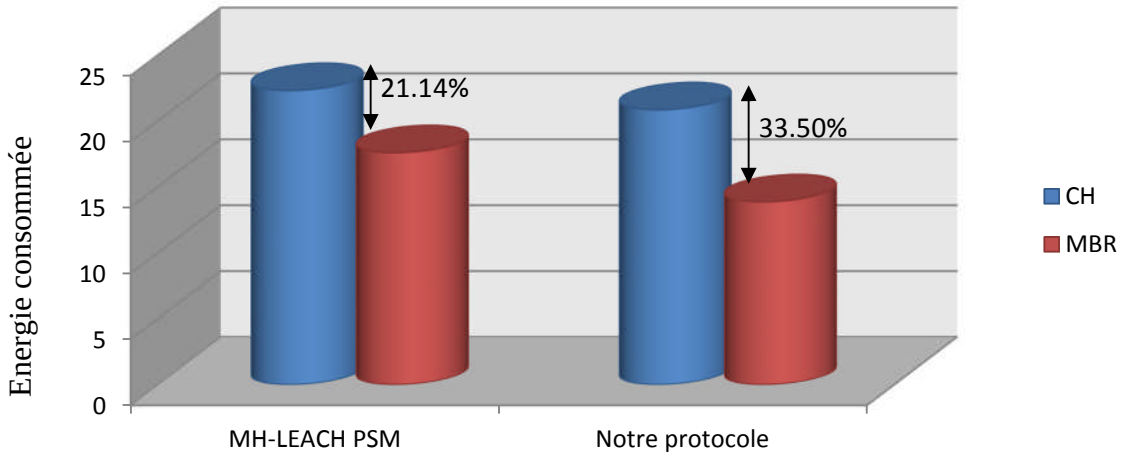
6.3.1 Consommation énergétique

Pour avoir les tests en consommation d'énergie, nous avons eu recours à l'onglet PowerProfiling de Tinyviz. Les cas de figure suivants peuvent se présenter :

a. **Consommation d'énergie des CH et des membres sur un échantillon de 20 nœuds**
Dans ce test, nous avons mesuré le taux de consommation d'énergie des CH par rapport aux nœuds membres pour les deux protocoles MH-LEACH PSM et notre solution.

	MH-LEACH PSM	Notre solution
Consommation énergétique des membres (Joules)	17.59	13.87
Consommation énergétique des CH (Joules)	22.32	20.86
Energie additionnelle des CH par rapport aux membres	21.14%	33.50%

Tableau IV.3 : Consommation d'énergie des CH par rapport aux membres.

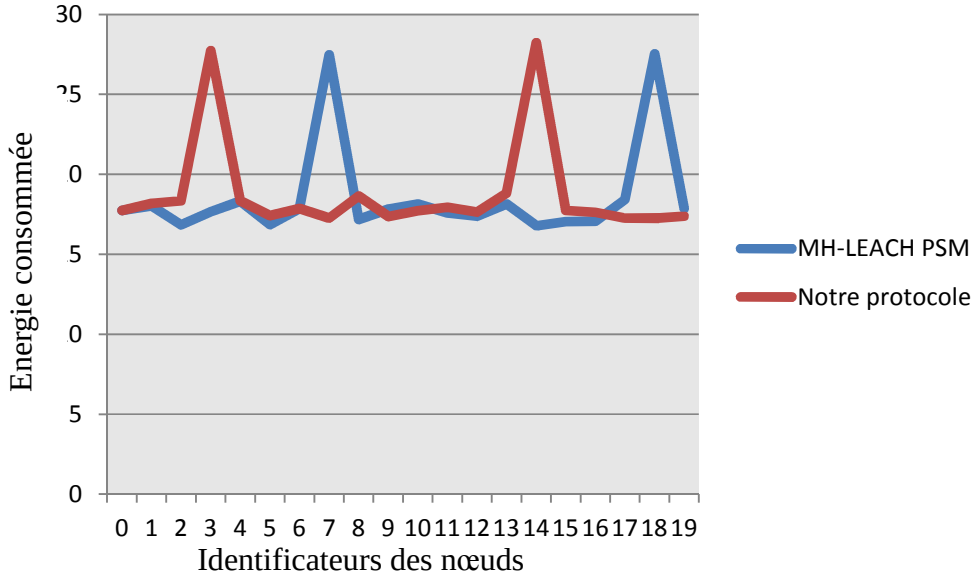


Energie moyenne consommée par les CH et les membres

Gph. IV.1 : Consommation d'énergie des CH par rapport aux membres.

Comme l'illustre le résultat du graphe VI-1-A, la moyenne de la consommation d'énergie des CH dans le protocole MH-LEACH PSM est plus élevée que celle des membres avec un taux de 36.02%. Cette hausse enregistrée dans la consommation d'énergie est induite par les tâches coûteuses en termes d'énergie qu'effectue le CH lors de son élection. Par contre, dans notre protocole, les nœuds effectuent, la vérification de services de sécurité à savoir le calcul des MAC. Dès lors, ce taux atteint 33.50% dans notre protocole sécurisé.
Par ailleurs, le graphe VI.1.B illustre que notre protocole maintient l'énergie additionnelle pour les CH par rapport aux membres. En effet, notre sécurisation ne surcharge pas les nœuds du réseau et n'inflige pas plus de tâches pour les CH par rapport aux membres.

b. **Consommation d'énergie par nœud sur un échantillon de 20 nœuds**



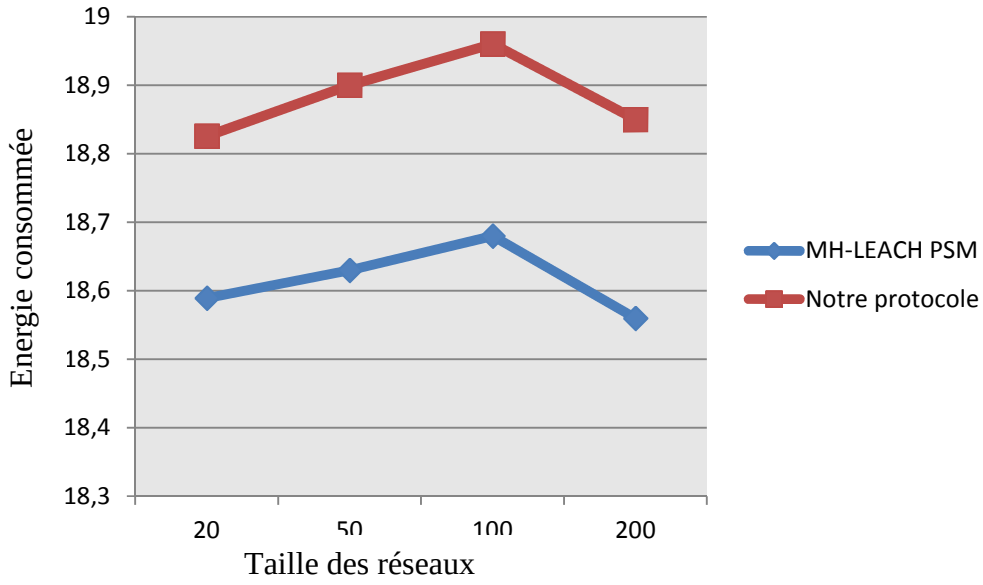
Gph. IV. 2 : Energie consommée par nœud.

Nous pouvons vérifier, en analysant le résultat du graphe VI-2, que les sommets représentent l'énergie consommée par des nœuds qui ont été élus CH durant la simulation.
Nous pouvons bien constater que les nœuds dans notre protocole consomment plus d'énergie que ceux du protocole MH-LEACH PSM, d'un taux approximativement égal à 1.27%, à cause de l'augmentation de messages de contrôle lors de notre sécurisation.

c. **Variation de la consommation d'énergie au nombre de nœuds du réseau**

Nombre de nœuds	20	50	100	200
Moyenne de consommation d'énergie dans MH-LEACH PSM (Joules)	18.58	18.63	18.42	18.56
Moyenne de consommation d'énergie dans notre protocole (Joules)	18.82	18.90	18.96	18.85

Tableau IV. 3 : Variation de consommation d'énergie au nombre de nœuds.

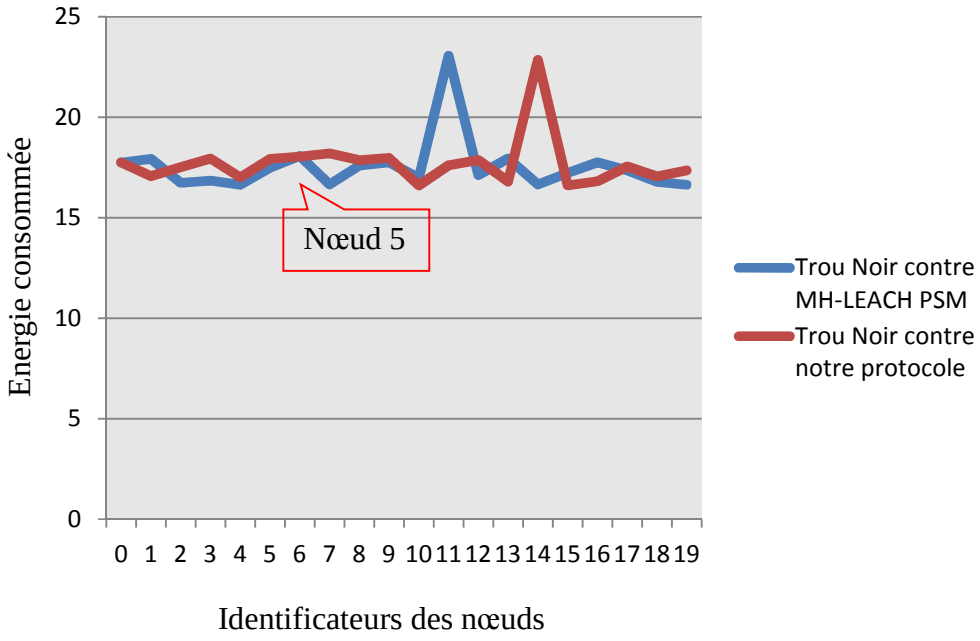


Gph. IV. 3 : Variation de consommation d'énergie au nombre de nœuds.

Comme l'illustre le résultat du graphe VI-3, nous remarquons que la moyenne d'énergie consommée dans le réseau est indépendante du nombre de nœuds déployés à cause de la topologie hiérarchique du protocole MH-LEACH PSM qui le rend très scalable . En effet, quand la taille du réseau augmente, le nombre de CH augmente. Donc, les nouveaux nœuds vont être affectés aux nouveaux CH et regroupés indépendamment des groupes déjà existants dans le réseau. Donc, malgré l'augmentation du nombre de nœuds déployés, la taille de tous les groupes est la même. Ainsi, tous les CH effectuent le même taux de tâches. Ainsi, MH-LEACH PSM maintient la consommation d'énergie des nœuds quelque soit la taille du réseau. Par ailleurs, nous constatons un taux de 1.76% d'énergie dissipée pour notre protocole par rapport au protocole MH- LEACH PSM. Cela revient à l'augmentation du nombre de messages de contrôle par les mécanismes de sécurisation.

6.4 Simulation de l'attaque Trou Noir

Dans cette partie, nous étudions les conséquences d'attaquer les deux protocoles MH-LEACH PSM et notre protocole sécurisé par Trou Noir. On procède à un test, le résultat du premier test concerne la dissipation d'énergie des nœuds. En effet, Trou Noir permet de supprimer tous les paquets de données qui transitent à travers lui.



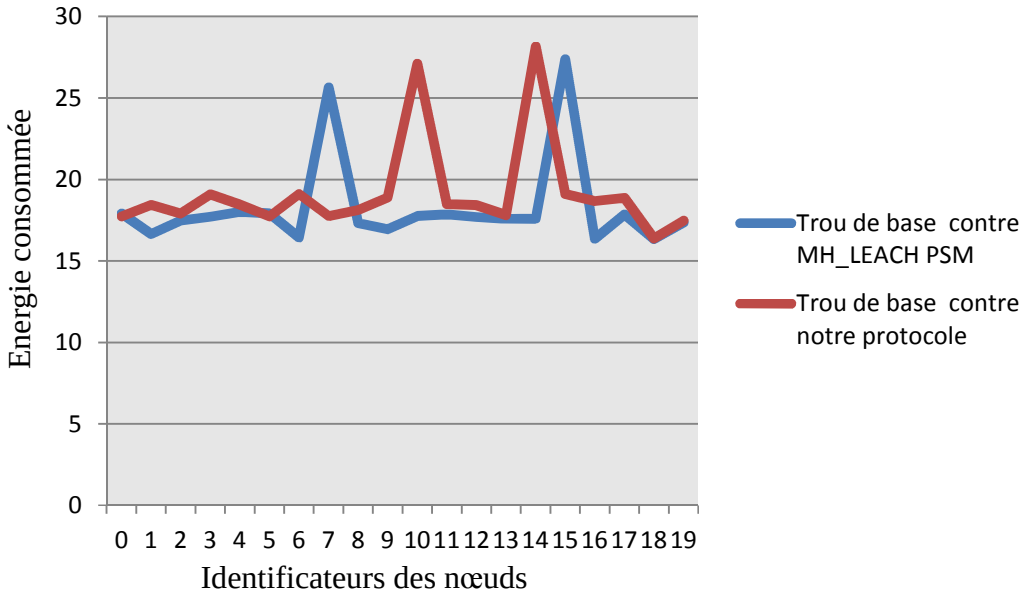
Gph. IV. 4 : Energie consommée par nœud.

Comme l'illustre le graphe IV.4 nous pouvons voir que le nœud 5 étant un chef de groupe ne consomme pas beaucoup d'énergie par rapport à l'autre CH par ce que ce nœud est un nœud attaquant et il n'agrège pas les donnée donc il ne consomme aucune énergie supplémentaire.

6.5 Simulation de l'attaque Trou de Base

Dans cette partie, nous étudions les conséquences de l'attaque de type Trou de Base sur les deux protocoles (MH-LEACH PSM et notre protocole) afin de comparer leurs consommations énergétiques.

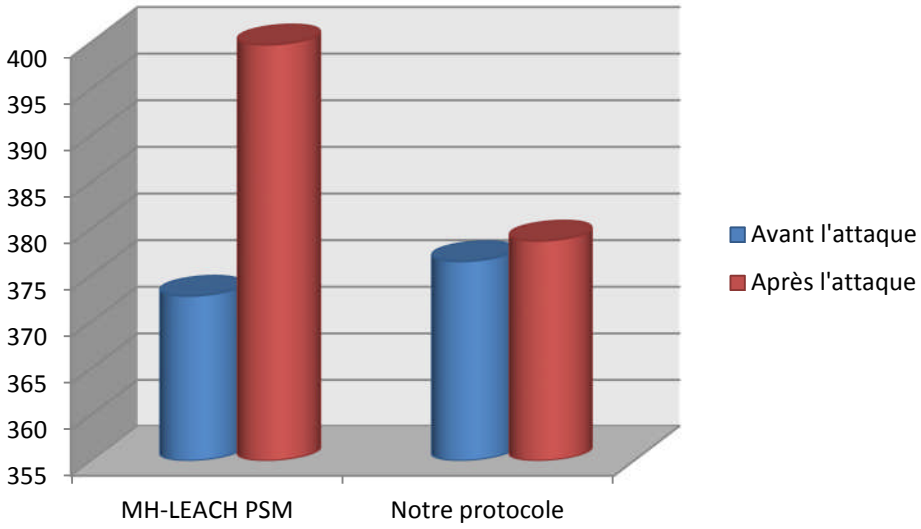
5.4.1. Consommation d'énergie par nœud



Gph. IV. 5 : Energie consommée par nœud.

Nous pouvons bien constater ,en analysant le résultat du graphe VI-5, que les nœuds dans notre protocole consomment plus d'énergie que ceux du protocole MH-LEACH PSM, d'un taux approximativement égal à 1.72%, à cause de l'augmentation de messages de contrôle lors de notre sécurisation.

5.4.2. La consommation d'énergie avant et après l'attaque Trou de Base



Gph. IV. 6 : La consommation d'énergie avant et après l'attaque Trou de Base

Comme l'illustre le graphe IV.6, nous pouvons apercevoir clairement la stabilité de la consommation d'énergie approximativement, avant et après l'attaque, car les nœuds ignorent le contenu des messages de l'attaquant grâce au service d'authentification utilisé, d'autre part, on remarque l'effet de cette attaque sur le protocole MH-LEACH PSM qui ne contient aucun mécanisme de sécurité traduite par une augmentation de la consommation d'énergie due aux

6. Conclusion

Dans ce chapitre, nous avons présenté l'implémentation ainsi que l'évaluation des deux protocoles MH-LEACH PSM et notre protocole sécurisé. Le système d'exploitation TinyOS est utilisé. Il consiste une programmation entière en langage NesC et une simulation avec TOSSIM .Nous avons réalisé une étude comparative entre notre protocole sécurisé et le protocole MH-LEACH PSM. Par la suite, nous avons implémenté l'attaque Trou Noir et l'attaque trou de base sur ce dernier afin de voir les effets néfastes que l'absence de sécurité peut donner. Cependant, ces attaques sont détectées par notre protocole grâce aux services de sécurité qu'il offre. Par ailleurs, nous avons constaté que les tests de performances effectués sur la consommation d'énergie ont montré que notre protocole répond bien aux critères de performances souhaités. Ainsi ces résultats nous ont permis de montrer l'efficacité de notre protocole en présence de l'attaque Trou Noir et de l'attaque trou de base. En effet, ce protocole ne permet pas de surcharger les nœuds capteurs, ni de dégrader les performances du réseau.

Conclusion générale

Les RCSF constituent des sujets de recherche innovants pour diverses disciplines des sciences et techniques de l'information et de la communication, mais avec toutefois des contraintes spécifiques s'érigeant en défis certains à relever. Durant notre travail de recherche, nous avons cerné ces problèmes qui se posent à l'heure actuelle dans ce type de réseaux. La sécurité en est un véritable enjeu, auquel une solution adéquate doit être apportée. Pour ce faire, on a étudié profondément la notion de sécurité dans les RCSF, aussi importante que primordiale pour la fiabilité du réseau. On a présenté des solutions efficaces et peu couteuses en énergie.

Plusieurs protocoles de routage et de sécurité sont présentés, et plusieurs classifications ont été établies. Nous étions intéressés très particulièrement par les protocoles de routage hiérarchiques pour leur gestion du réseau d'une manière à minimiser l'énergie consommée et le nombre de paquets de données échangés. Nous avons donc choisi le protocole de routage LEACH et sa variante MH-LEACH PSM, sur laquelle nous avons appliqué une taxonomie d'attaques et de solutions de sécurité.

Nous nous sommes intéressés à la sécurisation du service de routage qui est l'un des services piliers sur lesquels se base le fonctionnement d'un RCSF. Pour cela, il nous a fallu étudier les Protocoles LEACH et MH-LEACH PSM et les différentes attaques qui les ciblent, principalement les deux attaque trou noir et trou de base. Cette étude nous a permis de mettre en place une solution de sécurisation pour MH-LEACH PSM afin de répondre à la problématique de la détection des trous noirs et des trous de base afin de les écarter définitivement du réseau, dans le cas du trou noir nous avons vu que la manière la plus adéquate est de mettre en place une surveillance du comportement des nœuds CH pour détecter l'attaque et de mettre le nœud suspect en quarantaine et dans l'autre cas des trous de base on vérifie l'authentification des nœuds grâce au calcul des codes MAC.

Nous avons implémenté en nesC notre protocole sécurisé à l'aide du simulateur TOSSIM (et son interface graphique TinyVIZ) qui font partie de l'environnement de développement TinyOS. Ensuite, nous l'avons comparé au protocole MH-LEACH PSM. Les résultats indiquent que les performances du réseau n'ont pas été dégradées après la sécurisation concernant la consommation énergétique.

Pour la solution de routage que nous avons proposée, elle demeure intéressante dans la théorie mais nécessite tout de même des travaux d'expérimentation pour l'améliorer et démontrer son application pour les réseaux de capteurs sans fil. En guise de perspectives nous proposons de les appliquer dans un environnement réel composé de différents types de capteurs existants.

Notre travail s'est focalisé principalement sur une attaque spécifique nommé Trou Noir. Par ailleurs LEACH reste un protocole de routage très vulnérable face à d'autres attaques. Une des perspectives envisagées pour ce travail est de développer des mécanismes de sécurité pour prendre en charge des attaques plus avancées telles que brouillage radio (jamming), wormhole…

Il serait également plus intéressant d'implémenter ou d'utiliser d'autres protocoles d'authentification et de gestion de clés plus avancés pour sécuriser LEACH et ces variantes.

[1] C. D. Faundez, «**Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'´energie** », Thèse de Doctorat, Université Henri Poincaré, Nancy 1,2009.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, « **A Survey on Sensor Networks** », IEEE Communications Magazine, 2002.

[3] Q. Monnet, « **Modèles et mécanismes pour la protection contre les attaques par déni de service dans les réseaux de capteurs sans fil** », thèse de doctorat, Paris-Est, 2015.

[4] M.REMDHANI, « **Problèmes de sécurité dans les réseaux de capteurs avec prise en charge de l'énergie** » Mémoire de magister, université de Blida, 2013.

[5] W. ZNAIDI « **Quelques propositions de solutions pour la sécurité des réseaux de capteurs sans fil** », thèse de doctorat, Ecole doctorale : Informatique et Mathématiques de Lyon, 2010.

[6] J.M. Kahn, R.H. Katz, and K.SJ Pister. Next century challenges : « **mobile networking for smart dust. In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking**», pages 271–278. AC 1999.

[7] A-B Garcia-Hernando, J-F Martinez-Ortega, J-M Lopez-Navarro, A. Prayati, and L. Redondo-Lopez. « **Problem Solving for Wireless Sensor Networks**» , Springer-Verlag London, 2008.

[8] D-E Boubiche, « **Une approche Inter-Couches (cross-layer) pour la Sécurité dans les R.C.S.F** », thèse de doctorat, Université de Batna

[9] W. Bechkit, « **Un nouveau protocole de routage avec conservation d'énergie dans les réseaux de capteurs sans fil** », Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, 2009.

[10] F. Z. Benhamida,« **La tolérance aux pannes dans les réseaux de capteurs sans fil** », Rapport du mini projet, Institut National de Formation en Informatique INI, Algérie, 2006.

[11] M. OUABDESSELAM, « **Routage à économie d'énergie dans les réseaux de capteurs sans fil.** », Mémoire de Master2, UMMTO, 2012

[12] I.F. Akyildiz and M. C.VURAN, « **Wireless Sensor Networks**», John Wiley & Sons Ltd, 2010.

[13] E. Souto, R. Gomes, D. Sadok and J. Kelner, « **Sampling Energy Consumption in Wireless Sensor Networks**». IEEE International Conference on Sensor Networks,Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06), 2006.

[14] J. Cecilio, J. Costa and P. Furtado, «**Survey on Data Routing in Wireless Sensor Networks**» Springer-Verlag Berlin Heidelberg, 2010.

[15] I. Mahgoub and J. Ibriq, «**Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges**», International Symposium on Performance Evaluation of Computer and Telecommunication Systems' 2004 (SPECTS' 04), Page(s):759-769, California University, 2004.

[16] R Kaur, D Sharma and N Kaur, « **Comparative Analysis Of Leach And Its Descendant Protocols In Wireless Sensor Network**» , International Journal of P2P Network Trends and Technology, Vol. 3, Issue 1 pp. 51-55, 2013

[18] C.Y. Chong et S.P. Kumar, « **Sensor networks: Evolution, opportunities, and challenges**», Proceedings of the IEEE, vol. 91, n.8, 2003.

[19] T. Zhao, W. D. Cai, et Y. J. Li, « **A Sensor Network Topology Inference Algorithm omputational Intelligence and Security**».International Conference on. 2008.

[20] Jaap C. Haartsen, « **The Bluetooth radio system**». IEEE Personal Communications Magazine, 2000.

[21] Practel, Inc. ZigBee, « **Technology for Wireless Sensor Networks**». 2006.

[22] W. Bechkit, « **Un nouveau protocole de routage avec conservation d'énergie dans les réseaux de capteurs sans fil** », Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, 2009.

[23] A Amziane et M .Toumi « **Plateforme d'évaluation de la tolérance aux pannes des protocoles de routage dans les réseaux de capteurs sans fil**», Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, 2011.

[24] J.N. Al-Karaki and A. E. Kamel, «**Routing Techniques in Wireless Sensor Networks: A Survey** », Magazine: IEEE Communications, vol. 11, N° 6, pp. 6-28, 2004.

[25] R. Jurdak « **Wireless Ad hoc and sensor Networks: A Cross-Layer Design Perspective**» University college Dublin, 2007.

[26] D. Niculescu, «**Topics In Ad-Hoc Networks: Communication paradigms for Sensor Networks**», NEC Laboratories America, IEEE Communications Magazine, 2005.

[27] Y.Wang, C.Hsiao Tsai and H.Mao, « **HMRP: Hierarchy-Based Multipath Routing Protocol for Wireless Sensor Networks**» , Tamkang Journal of Science and Engineering, Vol. 9, No 3, pp. 255-264, 2006.

[28] S. Bandyopadhyay, E. Coyle, « **An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks**» , Proceedings of IEEE INFOCOM, Vol. 3, pp. 1713-1723. 2003.

[29] A. Bharathidasan and V. Anad Sau Ponduru, « **Sensor networks : An overview** », département d'informatique de Californie.

[30] M. LEHSAINI, « **Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique**», Thèse de Doctorat, 2009.

[31] I. Doghri, « **Stratégies de routage multi-chemin dans les réseaux sans fil multi-sauts** », ENS LYON, 2012

[32] E. DHIB ,« **Routage avec QoS temps réel dans les réseaux de Capteurs** »Ingénieur en Télécommunications option : Ingénierie des réseaux, école supérieure de communication de Tunis, 2006.

[33] M. Y. Romdhane, « **Evaluation des performances des protocoles S-MAC et Directed Diffusion dans les réseaux de capteurs** », Projet De Fin d'Etudes, école supérieur de communication de Tunis, 2007.

[34] M. CHARIF et A. BENYAGOUB « **Sécurisation du protocole de routage hiérarchique LEACH dans les réseaux de capteurs sans fil** », Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique, université Tlemcen, 2013

[35] M. Malik and Y. Singh, «**Analysis of LEACH Protocol in Wireless Sensor Networks**», International Journal of Advanced Research in Computer Science and Software Engineering, 2013.

[36] T. Murata and H. Ishibuchi, « **Performance evaluation of genetic algorithms for flowshop scheduling problems**», In Proceedings of the 1st IEEE Conference. Evolutionary Computation, volume 2, pp. 812–817. 1994.

[37] E. Lawrey, «**The suitability of OFDM as a modulation technique for wireless telecommunications, with a CDMA comparison**», Projet d'ingéniorat, Université James Cook, Australie, 2001.

[38] I. Guérin Lassous, « **Autonomic Computing : Accès au médium radio** », Cours M2 Recherche RTS, RTS5, Page(s) : 43-95, Université de Lyon, 2007.

[39] P. Radhakrishnan, « **Enhanced routing protocol for graceful degradation in wireless sensor networks during attacks**», Thèse d'ingéniorat, Université de Madras, Chennai, 2005.

[40] S-E BENBRAHIM, , « **défense contre l'attaque d'analyse de trafic dans les réseaux de capteurs sans fil** », Ecole polytechnique de Montréal, 2011

[41]W. Znaidi, M. Minier, J.P. Babau, « **An Ontology for Attacks in Wireless Sensor Networks**», Research Report RR-6704, INRIA. 2008.

[42] B.A Bensaber, « **Introduction à la sécurité des réseaux de capteurs sans fil** », Ecole d'été Internationale, Réseaux de Capteurs : Impacts et défis pour la sociétéUniversité de Bejaïa

[43] D. Martins, H. Guyennet, « **Etat de l'art. Sécurité dans les réseaux de capteurs sans fil**». Manuscrit auteur, publié dans SAR-SSI 2008 : 3rd conference on security of network architectures and information systems, France. 2008.

[44] M. SAXENA:« **Security in Wireless Sensor Networks: A Layer based Classification**», Purdue University, 2007.

[45] A. PERRIG, J. STANKOVIC, D. WAGNER: **« Security in Wireless Sensor Networks**», In Communications of the ACM, Vol. 47, No. 6. 2004.

[46] H. Bettahar, Y. Challal, «**Introduction à la sécurité informatique**», Supports de cours, Systèmes Intelligents pour le Transport, Université de Technologie de Compiègne, France, 2008.

[47] E. Bresson, «**Cryptographie: chiffrement par flot**», Séminaire de la cryptographie, Page(s): 22-34, Laboratoire de cryptographie, Université de Paris XII, 2001/2002.

[48] National Institute of Standards and Technology, « **Data Encryption Standard**» .In FIPS Publication 46-2. 1993.

[49] Nist Publication, « **The Advanced Encryption Standard (AES**) ». 2001.

[50] D. Baker, H.X. Mel, «**La cryptographie décryptée**», Livre, Nombre de Pages: 413, Edition Campus Press, 2001.

[51] A. BERRACHEDI et A. DIARBAKIRLI ,**« Sécurisation du protocole de routage hiérarchique LEACH dans les réseaux de capteurs sans fil** », Ecole nationale Supérieure d'Informatique (E.S.I), promotion 2008/2009.

[52] V.B RAJASHRE, V.C .PATIL, Dr. S.R. SAWANT et Dr. R.R. MUDHOLKAR, « **Classification and comparison of routing protocols in wireless sensor network**», Ubiquitous Computing and Communication Journal, Special Issue on Ubiquitous Computing Security Systems, Volume: Ubiquitous Computing Security Systems, 2009.

[53] **http:// www.zigbee.org**

[54] R. ANDERSON, H. CHAN et A. PERRING, «**Key Infection: Smart Trust for Smart Dust** » **,** IEEE International Conference, 2004.

[55] J. Ibriq and I. Mahgoub**.** « **A secure hierarchical routing protocol for wireless sensor networks**». In In: Proc. 10 IEEE International Conference on Communication Systems, pages 1–U″ 6, Singapore, 2006.

[56] C. Hong-bing, Y. Geng, and H. Su-jun. Nhrpa « **a novel hierarchical routing protocol algorithm for wireless sensor networks**» . The Journal of China Universities of Posts and Telecommunications, pages 75–81, 2008.

[57] Z. Quan and J. Li. « **Secure routing protocol cluster-gene-based for wireless sensor networks».** In Proc. The 1st International Conference on Information Science and Engineering (ICISE2009), pages 4098–4102, 2009.

[58] B. Parno, M. Luk, E. Gaustad, and A. Perrig. Lha-sp: « **secure protocols for hierarchical wireless sensor networks**» . In Proc. of 9th IFIP/IEEE International Symposium on Integrated Network Management, pages 31–44, 2005.

[59] M. Tubaishat, J. Yin, B. Panja, and S. Madria « **A secure hierarchical model for sensor network**». ACM SIGMOD Record, 33(1):7–13, 2004.

[60] S. Tixeuil, T. Herman, « **Un algorithme TDMA réparti pour les réseaux de capteurs** », INRIA Projet Grand Large, Universités Iowa et Paris-Sud XI, 2004.

[61] L.B. Oliveira, Hao C. Wong, M. Bern, R. Dahab, A. A. F. Loureiro. « **SecLEACH – A Random Key Distribution Solution for Securing Clustered Sensor Networks**». Fifth IEEE International Symposium on Network Computing and Applications (NCA'06).

[62] C. Karlof, N. Sastry, and D. Wagner. «**TinySec: A Link Layer Security Architecture for Wireless Sensor Networks** » 2004 Conference on Embedded Networked Sensor Systems Proceedings of the 2nd international conference on Embedded networked sensor systems.

[63] A. Braman and G.R Umapathi, « **A Comparative Study on Advances in LEACH Routing Protocol for Wireless Sensor Networks: A survey** » International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol. 3, issue 2, pp. 5683-5690, 2014.

[64] V. Loscri, G. Morabito, and S. Marano, « **A Two-Levels Hierarchy for Low-Energy Adaptive Clustering Hierarchy (TL-LEACH)**» in 62nd Vehicular Technology Conference, IEEE, vol.3, 2005, pp.1809-1813.

[65] N. Sindhwani and R. Vaid, «**V LEACH: AN Energy Efficient Communication Protocol for WSN** » Mechanica Confab, vol. 2, no. 2, pp. 79-84, 2013.

.

[66] J. Gnanamdigai, Dr. N. Rengarajan, and K. Anbukkarasi, « **Leach and Its Descendants Protocols : A Survey**, » Intrernationnal Journal of Communication and Computer Technlgies (IJCCT), vol. 01, issue 02, no. 3, pp 15-21, 2012.

[67] M. Usha and Dr. N. Sankarram, « **A Survey on Energy Efficient Hierarchical (Leach) Clustering Algorithms in Wireless Sensor Network**, » International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14), vol.2, special issue 1, pp. 601-609, 2014.

[68] A. Depedri, A. Zanella and R. Verdone, « **An Energy Efficient Protocol for Wireless Sensor Networks**» in Proc, pp. 1-6, AINS, 2003.

[69] R. Kaur, D. Sharma, and N. Kaur, « **Comparative Analysis Of Leach And Its Descendant Protocols In Wireless Sensor Network,** » International Journal of P2P Network Trends and Technology (IJPNTT), vol. 3, issue 1, pp. 51-55, 2013.

[70] V. Kumar, S. Jain, and S. Tiwari, « **Energy Efficient Clustering Algorithms in Wireless Sensor Networks: A Survey**, » International Journal of Computer Science Issues (IJCSI), vol. 8, issue 5, no. 2, pp. 259-268, 2011.

[71] A. Yektaparast and F.H. Nabavi, and A. Sarmast, « **An Improvement on LEACH protocol (Cell-LEACH**),» in 14th International Conference on Advanced Communication Technology (ICACT), pp.992-996, 19-22, 2012.

[72] E. Abdellah, and S. Benalla, A.B. Hsaane, and M. Lahcen, « **Advanced Low Energy Adaptive Clustering Hierarchy**, » (IJCSE) International Journal on Computer Science and Engineering, vol. 02, no. 07, pp. 2491-2497, 2010.

[73] P. Manimala and R. Senthamil selvi, « **A Survey on Leach-Energy Based Routing Protocol**, » International Journal of Emerging Technology and Advanced Engineering (IJETAE), vol.3, issue 12, pp. 657-660, 2013.

[74] Ch. KARLOF and D. WAGNER, « **Secure routing in wireless sensor networks: attacks and countermeasures**, » Ad Hoc Networks, 293–315, 2003.

[75] T. Roosta and S. W. Shieh and S. S. Sastry, « **Taxonomy of security attacks in sensor networks**, » in: First IEEE International Conference on System Integration and Reliability Improvements, IEEE Computer Society, 2006.

[76]] J. Cai, P. Yi, J. Chen, Z. Wang, N. Liu, « **An adaptive approach to detecting black and gray hole attacks in ad hoc network**, » in: 4th IEEE International Conference on

Advanced Information networking and Applications, IEEE Computer Society, pp. 775–780. 2010.

[77] A. C. FERREIRA, M. A. VILACA, L. B. OLIVEIRA, E. HABIB, H. C.WONG, et A. A. LOUREIRO, « **On the security of cluster-based communication protocols for wireless sensor networks**». In Proc. 4th IEEE International Conference on Networking (ICNŠ05), volume 3420 of Lecture Notes in Computer Science, pages 449–458, 2005.

[78] L. B. OLIVERIA, H. C. WONG, M. BERN, R. DAHAB, et A. A. F. LOUREIRO. « **SecLEACH - a random key distribution solution for securing clustered sensor networks. In Proc. of the Fifth IEEE International Symposium on Network Computing and Applications**, » pages 145–154, Washington, DC, USA, IEEE Computer Society. 2006.

[79] L. B. OLIVEIRA, A. FERREIRA, M. A. VILACA, H. C. WONG, M. BERN, R. DAHAB, et A. A. F. LOUREIRO, « **SecLEACH-on the security of clustered sensor networks**». Signal Processing, 87(12):2882–2895, 2007.

[80] E. Lawrey, «**The suitability of OFDM as a modulation technique for wireless telecommunications, with a CDMA comparison**», Projet d'ingéniorat, Université James Cook, Australie, 2001.

[81] M. ABED, M. BACHA, «**Description des comportements d'un réseau de capteurs Sans fils à l'aide de SMA, »** mémoire de fin d'études, Ecole nationale Supérieure d'Informatique ESI, Algérie, 2012.

[82] S. Tixier, « **TinyOS** », Mini rapport, LIF12, Université Lyon 1, 2007.

[83] M. Badnet, N. Belloir «**Réseaux de capteurs : Mise en place d'une plateforme de test et d'expérimentation** », Master Technologie de l'Internet 1ère année, France, 2005.

[84] W. Znaidi, « **Modélisation formelle de réseaux de capteurs à partir de TinyOS** », Projet de fin d'études, Ecole Polytechnique de Tunisie, 2006.

[85] H. Alatrista, J. Mathieu, K. Gouaïch S. Aliaga, « **Implémentation de protocoles sur une plateforme de réseaux de capteurs sans fil** », TER master 1 informatique, Université de Montpellier II, 2008.

[86] B. Chen, G. Werner Allen, M. Hempstead, M. Welsh, V. Shnayder, « **Simulating the Power Consumption of LargeScale Sensor Network Applications**», Proceedings of the 2nd international conference on Embedded networked sensor systems, Pages: 188 – 200, Harvard University, 2004.

[87] H. Sundani, H. Li, V. Devabhaktuni, M. Alam,P. Bhattacharya, « **Wireless Sensor Network Simulators : A Survey and Comparisons**». International Journal Of Computer Networks (IJCN), Vol. 2 (5). 2010.

[88] A. Dunkels, B. Grönvall, T. Voigt, « **Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors** », 29th Annual IEEE International Conference on Local Computer Networks, Pages: 455–462, Swedish Institute of Computer Science, 2004.

[89] C. Han, E. Kohler, M. Srivastava, R. Kumar, R. Shea, « **A Dynamic Operating System for Sensor Nodes**», Proceedings of the 3rd International Conference on Mobile Systems, Applications and Services (Mobisys), Page(s): 163-176, University of California, Los Angeles, 2005.

[90] C. Duffy, C. J. Sreenan, J. Herbert, U. Roedig, « **A Performance Analysis of MANTIS and TinyOS** », Technical Report CS-2006-27-11, University College Cork, Ireland, 2006.

[91] H. Alatrista, J. Mathieu, K. Gouaïch S. Aliaga, « **Implémentation de protocoles sur une plateforme de réseaux de capteurs sans fil** », TER master 1 informatique, Université de Montpellier II, 2008.

[92] D. Gay, P. Levis, « **TinyOS Programming** », Livre, ISBN: 0521896061, Nombre de Pages: 264, Presse de l'université de Cambridge, 2006.

[93] R. Rivest., « **The MD5 Message-Digest Algorithm**», RFC 1321.1992.

[94] D. Eastlake, P. Jones, « **US Secure Hash Algorithm 1 (SHA1**) » , RFC 3174. 2001.

[95] H. Hadjammar, N. Doufene, **« Routage dans les réseaux de capteurs : optimisation du protocole Directed Diffusion»,** Projet de fin d'étude, Institut National de formation en Informatique INI, Algérie, 2006.

[96] D MALKHI ET M. REITER **«Unreliable Intrusion Detection in Distributed Computation**s **»**. In Proc. 10 th Computer Security Foundations Workshop (CSFW97), 1997.

[97] C. Dwork, N.A. Lynch and L. Stockmeyer **«Consensus in the presence of partial synchrony »** Journal of the ACM, 1988.

Annexe A : Le système d'exploitation TinyOS

A.1. Présentation générale de TinyOS

TinyOS est un système d'exploitation open source conçu pour les réseaux de capteurs par l'université américaine de BERKELEY. Le caractère open source permet à ce système d'être régulièrement enrichi par une multitude d'utilisateurs. Sa conception a été entièrement réalisée en NesC, langage orienté composant syntaxiquement proche du C. Il respecte une architecture basée sur une association de composants, réduisant ainsi la taille du code nécessaire à sa mise en place. Cela s'inscrit dans le respect des contraintes de mémoires qu'observent les capteurs pourvus de ressources très limitées dues à leur miniaturisation.

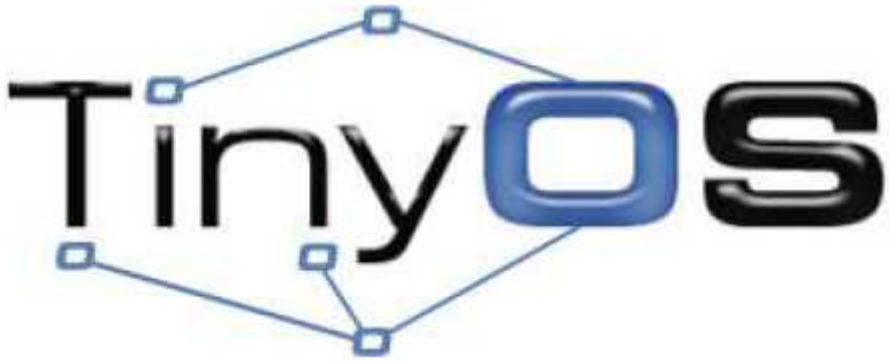


Fig. A-1 : Cigle du système d'exploitation TinyOS.

Pour autant, la bibliothèque de composants de TinyOS est particulièrement complète puisqu'on y retrouve des protocoles réseaux, des pilotes de capteurs et des outils d'acquisition de données. Un programme s'exécutant sur TinyOS est constitué d'une sélection de composants systèmes et de composants développés spécifiquement pour l'application laquelle il sera destiné (mesure de température, taux d'humidité…).
TinyOS s'appuie sur un fonctionnement évènementiel, c'est-à-dire qu'il ne devient actif qu'à l'apparition de certains évènements. Le reste du temps, le capteur se trouve en état de veille, vu les faibles ressources énergétiques des capteurs, garantissant ainsi une durée de vie maximale. Ce type de fonctionnement permet une meilleure adaptation à la nature aléatoire de la communication sans fil entre capteurs.

A.2. Caractéristiques de TinyOS

TinyOS a été créé pour répondre aux caractéristiques et aux nécessités des RCSF telles que :

· **Taille réduite** : TinyOS a une empreinte mémoire très faible puisqu'il ne prend que 4 Ko de mémoire libre et 300 à 400 octets dans le cadre d'une distribution minimale.

·**Applications orientées composants**: Un programme s'exécutant sur TinyOS est constitué d'une sélection de composants qui peut être utilisée telle quelle ou bien adaptée à une application précise (mesure de température, du taux d'humidité, etc.). A cette fin, TinyOS fournit une réserve de composants systèmes utilisables au besoin. Parmi les plus fréquents, on

cite ceux concernant les entrée/sorties, les timers, etc. TinyOS utilise un Langage de Description d'Architecture afin de définir quels sont les composants impliqués dans la création de l'application ainsi que la manière dont ils sont reliés. Cette liaison entre composants repose sur la notion d'interface.

· **Programmation orienté évènement**: Le plus gros avantage de TinyOS est qu'il est basé sur un fonctionnement événementiel, c'est à dire qu'il ne devient actif qu'à l'apparition de certains évènements. Le reste du temps, le capteur se trouve en état de veille afin de garantissent une durée de vie maximale aux faibles ressources énergétiques du capteur.Ce fonctionnement évènementiel (event-driven) s'oppose au fonctionnement dit temporel (time-driven) où les actions du système sont gérées par une horloge donnée.

· **Non Préemptif:** Le caractère préemptif d'un système d'exploitation précise si celui-ci permet l'interruption d'une tâche en cours. TinyOS ne gère pas ce mécanisme de préemption entre les tâches. Autrement dit, une tâche ne peut pas interrompre une autre tâche. Ce mode de fonctionnement permet de bannir les opérations pouvant bloquer le système et donne la priorité aux interruptions matérielles (i.e. les événements peuvent interrompre les tâches).
TinyOS est donc basé sur une structure à deux niveaux de planification :
- Les évènements : ils sont utilisés pour réaliser des processus urgents et courts.
- Les tâches : les tâches sont pensées pour réaliser une plus grande quantité de traitements et elles ne sont pas critiques dans le temps. Les tâches sont exécutées complètement, mais l'initialisation et la terminaison d'une tâche sont des fonctions séparées. Les tâches ne peuvent pas prendre de paramètre en entrée.

· **Pas de temps réel** : Lorsqu'un système est dit « temps réel » celui-ci gère des niveaux de priorité dans ses tâches permettant de respecter des échéances données par son environnement. Dans le cas d'un système strict, aucune échéance ne tolère de dépassement contrairement à un système temps réel mou. TinyOS se situe au-delà de ce second type car il n'est pas prévu pour avoir un fonctionnement temps réel.

A.3. Equipements supportés par TinyOS

TinyOS peut être implémenté sur un PC capteur (ATMega8, AVRMote, Mica, Rene2, MSP430, Telos). Au-delà de cette liste, il est possible d'implémenter tout type de plateforme embarquée physique en redéveloppant les bibliothèques nécessaires à la prise en compte des entrées sorties nécessaires. Citant comme exemple le résultat d'une thèse mettant en oeuvre TinyOS sur un dispositif Freescale MC13192-EVB (semi-conducteur utilisé pour évaluer des plateformes) sur un réseau ZigBee**.**

A.4. Allocation de la mémoire

Il est très important d'aborder la façon avec laquelle un système d'exploitation gère la mémoire, d'autant plus lorsque ce système travaille dans un environnement aussi restreint.

TinyOS occupe un espace mémoire faible répartie en :
·Pile: sert de mémoire temporaire au fonctionnement du système notamment pour l'empilement et le dépilement des variables locales.

· Variables globales : réservent un espace mémoire pour le stockage de valeurs pouvant être accessible depuis des applications différentes.

· Mémoire libre : pour le reste du stockage temporaire.

TinyOS possède une mémoire fixe. En effet, il interdit les allocations dynamiques ainsi que celles se produisant à l'exécution. De plus, les pointeurs de fonctions n'existent pas. Pour cela, TinyOS s'appuie sur le graphe de composants précédemment décrit afin de déterminer la taille de chaque composant et ainsi établir statiquement leurs liaisons à la compilation. Par ailleurs, il n'existe pas de mécanisme de protection de la mémoire sous TinyOS, ce qui rend le système particulièrement vulnérable aux corruptions de la mémoire.

A.5. Allocation de ressources

Le choix d'un ordonnanceur détermine le fonctionnement global du système et le dotera de propriétés précises telles que la capacité à fonctionner en évènementiel.
L'ordonnanceur TinyOS se compose de :
– 2 niveaux de priorités (bas pour les tâches, haut pour les évènements).
– 1 file d'attente FIFO (disposant une capacité de 7).
A l'appel d'une tâche, celle-ci va prendre place dans la FIFO en fonction de sa priorité (plus elle est grande, plus le placement est proche de la sortie). Dans le cas où la file d'attente est pleine, la tâche dont la priorité est la plus faible est enlevée de la file FIFO. Lorsque la file est vide, le système met en veille le dispositif jusqu'au lancement de la prochaine interruption.

Annexe B : Le langage de programmation NesC

B.1. Présentation générale de NesC

NesC est une extension du langage de programmation C. Il est conçu pour incarner les concepts structurant et le modèle d'exécution de TinyOS. Les composants sont les éléments de base pour former une application NesC. Chaque composant correspond à un élément matériel (LED, timer, ADC …) et peut être réutilisé dans différentes applications.
Les composants NesC fournissent ou utilisent des interfaces bidirectionnelles qui définissent d'une manière abstraite les interactions entre deux composants. L'utilisation des mots clés **use** et **provide** au début d'un composant permet de savoir respectivement si celui-ci fait appel à une fonction de l'interface ou redéfini son code. Il est à noter que tous les composants NesC doivent posséder l'interface StdControl car celle-ci est utilisée pour initialiser, démarrer et arrêter les composants.

Les composants NesC présentent des similarités avec des objets. Les états sont encapsulés et on peut y accéder par des interfaces. En NesC, l'ensemble des composants et leurs interactions sont fixés à la compilation pour plus d'efficacité. Ce type de compilation permet d'optimiser l'application pour une exécution plus performante. En langage objet, cette phase est réalisée lors de l'exécution ce qui rend celle-ci plus lente.

B.2. Implémentation d'une application NesC

Pour implémenter une application NesC, il faut avoir connaissance sur la structure et le fonctionnement des composants et des interfaces qui la constituent. Cette partie permet de bien expliquer ces notions. Il est néanmoins recommandé de faire recours aux leçons au niveau du tutorial TinyOS qui englobe tous les besoins de programmation NesC en accédant à **C:\tinyos\cygwin\opt\tinyos-1.x\doc\tutorial**

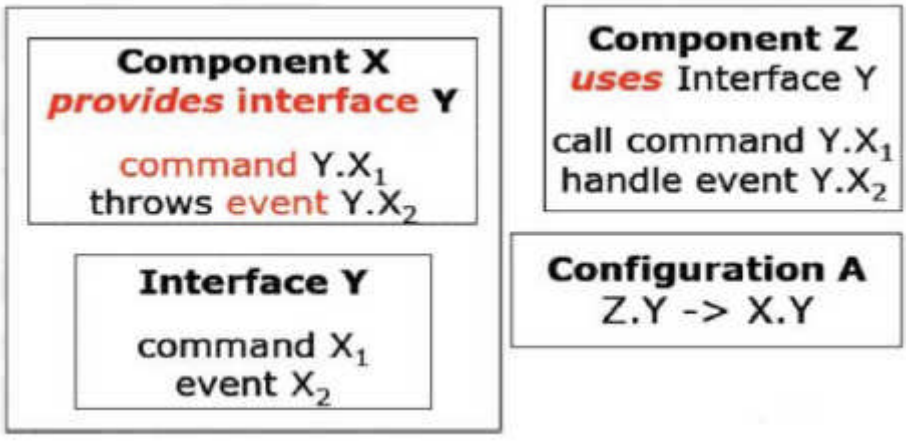


Fig. B-1 : Architecture générale d'une application NesC.

B.2.1. Les interfaces

Une interface déclare deux types de fonctions: commandes et événements. Afin de distinguer ces fonctions, leurs en-têtes sont précédés des mots-clés respectifs **event** ou **command**. Les commandes font typiquement des appels du haut vers le bas (des composants applicatifs vers les composants plus proches du matériel). Tandis que les événements remontent les signaux du bas vers le haut. Pour appeler une commande, il faut utiliser le mot-clé **call**.

Par exemple:

call Send.send (1, sizeof(Msg), &msg1) ;

Pour signaler un événement, il faut utiliser le mot-clé **signal**. Par exemple:

signal Send.sendDone(&msg1, SUCCESS);

Le modèle mémoire fixé par TinyOS n'autorise pas les pointeurs de fonctions. Afin de proposer un mécanisme alternatif, NesC utilise des interfaces paramétrées. Celles-ci permettent à l'utilisateur de créer un ensemble d'interfaces identiques et d'en sélectionner une seule à appeler grâce à un identifiant. Par exemple :

interface SendMsg **[uint8_t id]**

B.2.2. Les composants

Il existe deux types de composants : les configurations et les modules.

B.2.2.1. Les configurations

Elles permettent de décrire les composants composites, i.e., des composants composés d'autres composants. Elles relient les interfaces utilisées par certains composants aux interfaces offertes par d'autres composants. Une configuration est donc constituée de modules et/ou d'interfaces ainsi que de la description des liaisons entre ces composants. Il existe trois possibilités de connexion:

· End-point1 = End-point2
· End-point1 -> End-point2
· End-point1 <- End-point2 (équivalent à : endpoint2 -> endpoint1)

Les éléments connectés doivent être compatibles : Interface à interface, event à event, etc. Il faut toujours connecter un utilisateur d'une interface à un fournisseur de l'interface. Il est à noter que la configuration Main est obligatoirement présente dans la configuration décrivant l'ensemble de l'application car son rôle est de démarrer l'exécution de l'application.

B.2.2.2. Les modules

Ce sont les éléments de base de la programmation. Ils permettent de fournir les codes des applications NesC. Pa ailleurs, il est à noter que le modèle d'exécution proposé par NesC repose sur les tâches et les gestionnaires d'interruption. Donc, les modules permettent aussi d'implémenter ces tâches.

Une tâche est un ordonnancement FIFO utilisée pour réaliser un travail qui nécessite beaucoup de calculs. Elle peut être postée par une commande ou un événement. C'est un élément de contrôle indépendant défini par une fonction retournant **void** et sans arguments :

task void NomTask() { ... }

Les tâches sont lancées en les préfixant par **post**:

post NomTask();

B.3. Compilation d'une application NesC

Les fichiers de NesC portent l'extension .nc. Par ailleurs, le compilateur de NesC est appelé ncc. Pour effectuer la compilation, les fichiers sources doivent se situer dans le même répertoire contenant aussi un makefile de la forme :

COMPONENT= nom de l'application
include ../Makerules

Ce Makefile permet de compiler le composant en spécifiant en paramètre la plateforme sur laquelle doit fonctionner l'application. Par exemple, pour un capteur de type mica2, la commande permettant de compiler l'application sera : make mica2. Le compilateur ncc offre aussi la possibilité de pouvoir compiler l'application pour l'utiliser sur un simulateur de TinyOS. Dans ce cas, la commande sera : **make pc**. Cette commande génère un exécutable **main.exe** dans l'arborescence **/repertoire_courant/build/pc**.

B.4. Exemple illustratif d'une application NesC

On va donner l'exemple universel « Bonjour » ou « Hello » pour mieux illustrer la structure d'une application NesC

```
module HelloM {
provides {
interface StdControl;
}
uses {
interface Timer; interface Leds;
}
}
```

HelloM.nc (1)

```
implementation {
command result_t StdControl.init(){
call Leds.init();
return SUCCESS;
} }
command result_t StdControl.start(){
```

```
return call
Timer.start(TIMER_ONE_SHOT, 1000);
}
command result_t StdControl.stop(){
return call Timer.stop();
}                                HelloM.nc(2)
```

```
event event Timer.fired(){
call Leds.redOn();
call Leds.greenOn();
call Leds.yellowOn();
return SUCCESS;
}
} // implementation
                                 HelloM.nc(3)
```

```
configuration Hello {
}
implementation {
components Main, HelloM, SingleTimer, LedsC ;
Main.StdControl ---> HelloM.StdControl;
Main.StdControl – –> SingleTimer;
HelloM.Timer ---> SingleTimer.Timer ;
HelloM.Leds ---> LedsC ;
```

Fig. B-2 : Exemple illustratif d'une application NesC.

Annexe C : L'interface graphique TinyViz

C.1. Présentation générale de TinyViz

TinyViz est fourni avec TinyOS. Il s'agit d'une interface graphique programmée en langage JAVA. Elle permet de représenter un RCSF émulé grâce au simulateur TOSSIM. Pour plus d'informations sur l'utilisation de TinyViz, aller à:
http://www.tinyos.net/tinyos-1.x/doc/tutorial/lesson5.html

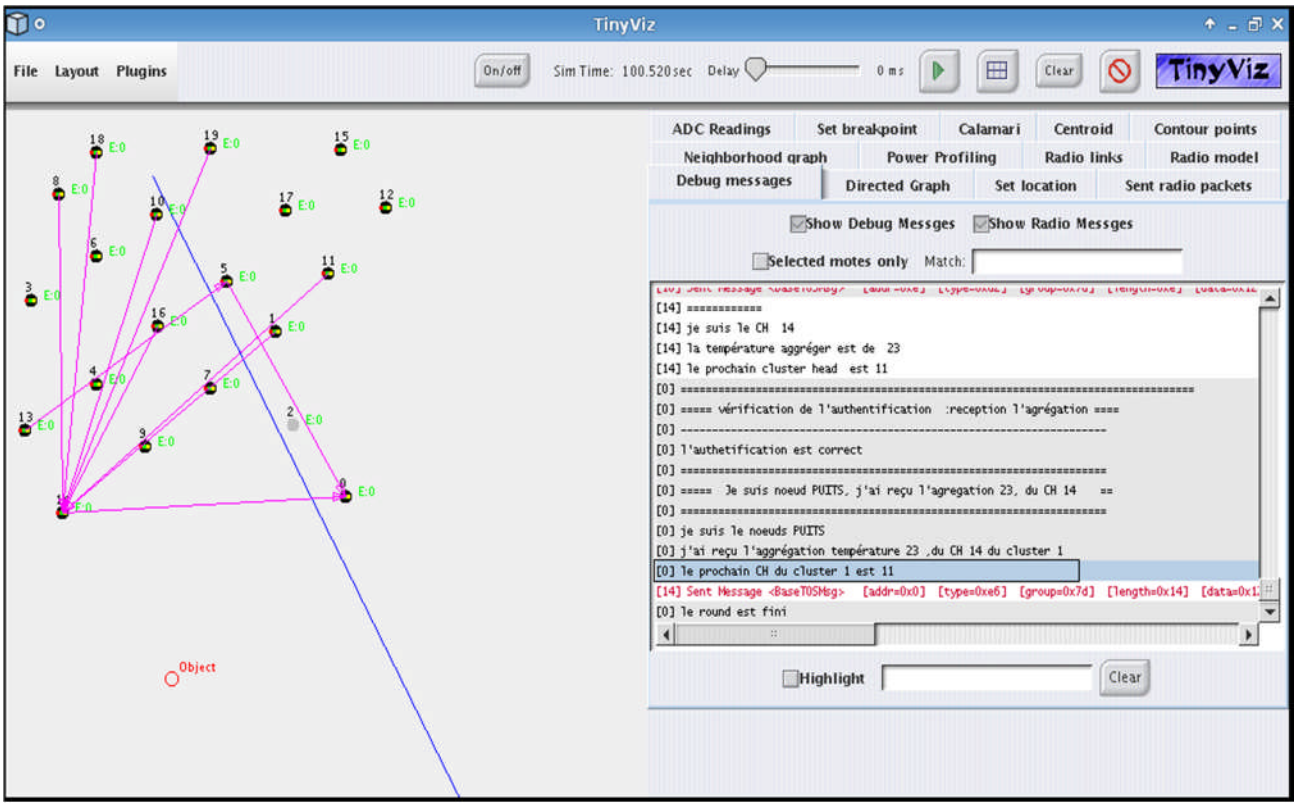


Fig. C-1: fenêtre graphique de TinyViz.

Une fois TinyViz est lancé, on peut visualiser une fenêtre comme celle illustrée dans la figure C.1. Dans la partie gauche de cette figure, on distingue les capteurs qui sont déplaçables dans l'espace. Quant à la partie droite, on distingue les commandes permettant d'intervenir sur la simulation:

· On/Off: met en marche ou éteint un capteur.
· Delay: permet de sélectionner la durée au bout de laquelle se déclenche le timer.
· Play: permet de lancer la simulation où de la mettre en pause
· Bouton de grilles: affiche un quadrillage sur la zone des capteurs afin de pouvoir les situer dans l'espace.
· Clear: efface tous les messages qui avaient été affichés lors de la simulation.
· Stop: arrête la simulation et ferme la fenêtre.

Pour lancer une application, il faut régler le **Delay** souhaité entre chaque application, choisir les plugins de visualisation que l'on souhaite, et, appuyer sur **Play**. La simulation démarre. Chaque onglet contient un plugin qui permet de visualiser la simulation de façon plus ou moins détaillée. Par exemple, en activant le plugin **Debug Messages**, tous les messages de type **Debug** apparaîtront dans l'onglet correspondant. Le plugin **Radio Links** permet de visualiser graphiquement par des flèches, les échanges effectués entre les capteurs. Plus précisément, si un capteur envoie un broadcast, il sera repéré par un cercle. Par contre, s'il envoie un message direct (unicast) alors le lien de communication sera repéré par une flèche.

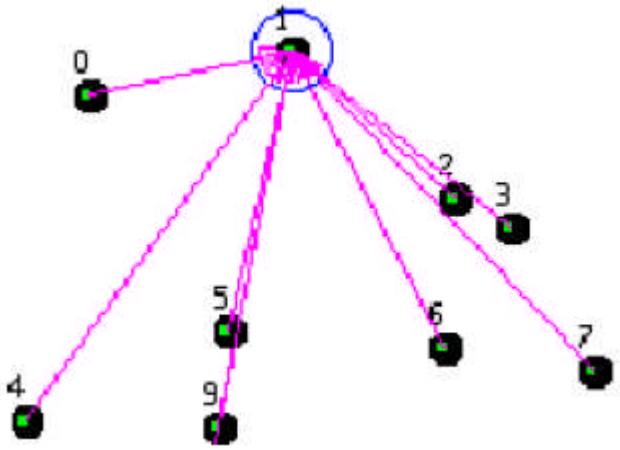


Fig. C-2 : Echange de messages entre les nœuds.

Annexe D: PowerTOSSIM

D.1. Présentation générale

Le simulateur TOSSIM n'a pas la capacité de vérifier le taux d'énergie dissipée pendant l'exécution des applications. Cependant, le besoin de vérifier la consommation énergétique dans un RCSF a un intérêt primordial. L'université de Harvard a conçu le simulateur PowerTOSSIM qui surmonte ce problème. Ce nouveau simulateur est intégré dans TOSSIM. Il permet de calculer le total d'énergie consommée par chaque composant constituant l'architecture de TOSSIM (LED, radio, CPU, etc.).
Pour simuler ces composants, on fait appel au module PowerState. Ce dernier engendre des messages de transition d'états d'énergie (*power state transition messages*) pour chaque composant. Ces messages peuvent être combinés avec un modèle d'énergie pour générer en détail les consommations d'énergie. Pour se faire, un fichier programmé en langage python, intitulé « postprocess.py » est utilisé.

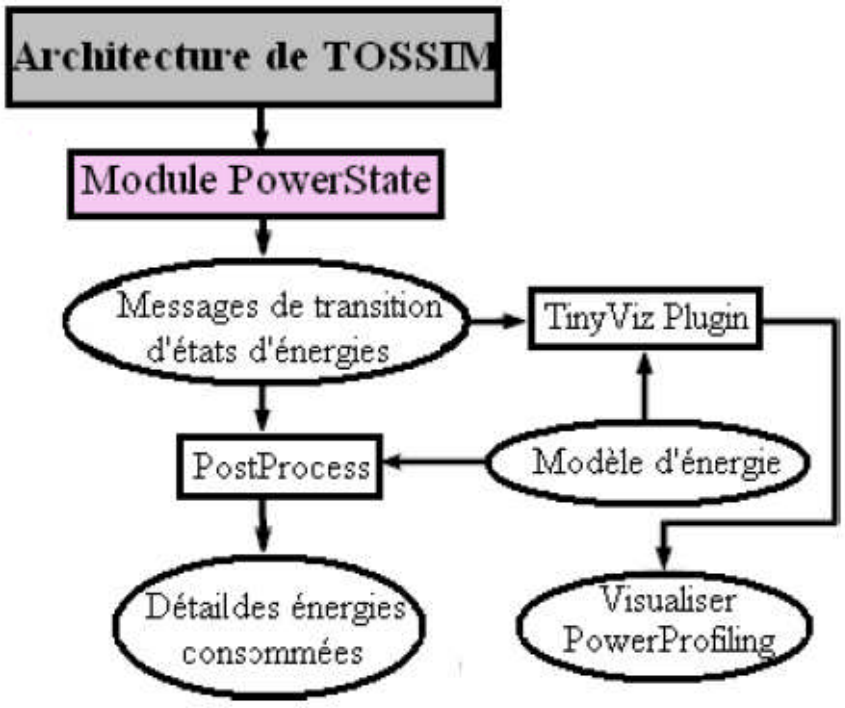


Fig. D-1: Architecture de PowerTOSSIM.

D.2. Lancer PowerTOSSIM

A- Pour récupérer l'énergie consommée par les nœuds du réseau, il faut passer par ces étapes:

1- Accéder à l'application à simuler et la compiler en tapant: **make pc**

2- Taper **export DBG=power**
3- Exécuter **main.exe** en choisissant le temps de simulation avec **–t** et le nombre de noeuds du réseau avec **–p**. Une trace de simulation est enregistré dans un fichier dont l'extension est .trace .Pour se faire, taper **: /build/pc/main.exe -t=60 -p 10 > NomApp.trace**
(Le temps est égal à 60 secondes et le nombre de noeuds à 10)
4- Exécuter **postprocess.py** sur la trace de simulation en spécifiant les paramètres –sb et --em **/opt/tools/scripts/PowerTOSSIM/postprocess.py –sb=0 –em /opt/tools/scripts/Power TOSSIM/mica2_energy_model.txt NomApp.trace**
Le paramètre –sb spécifie si les nœuds sont attachés à un autre nœud (i.e. embarqué). En Outre, le paramètre –em spécifie le modèle d'énergie. Pour plus de détail sur l'utilisation d'autres paramètres de PowerTOSSIM, exécuter **postprocess.py --help**

5- Le résultat enregistre l'énergie totale utilisée par chaque composant sur chaque noeud. Il est sous la forme suivante :
Mote 0, cpu total: 719.503906
Mote 0, radio total: 1235.255862
Mote 0, adc total: 0.000000
Mote 0, leds total: 571.570576
Mote 0, sensor total: 0.000000
Mote 0, eeprom total: 0.000000
Mote 0, cpu_cycle total: 0.000000
Mote 0, Total energy: 2526.330344
.
.
Mote 9, cpu total: 635.394462
Mote 9, radio total: 1090.990102
Mote 9, adc total: 0.000000
Mote 9, leds total: 504.416514
Mote 9, sensor total: 0.000000
Mote 9, eeprom total: 0.000000
Mote 9, cpu_cycle total: 0.000000
Mote 9, Total energy: 2230.801078

6- Pour ne pas perdre ce résultat, il est commode de le sauvegarder dans un fichier texte. Pour se faire, Ajouter dans l'instruction de l'étape 4:
/opt/tools/scripts/PowerTOSSIM/postprocess.py –sb=0 –em /opt/tools/scripts/Power TOSSIM/mica2_energy_model.txt NomApp.trace > Result.txt

7- Pour avoir un résultat d'énergie plus détaillé, ajouter le paramètre **–detail** dans l'instruction de l'étape 4. Le résultat est enregistré automatiquement dans des fichiers textes dont le nombre est égal au nombre de noeuds simulés. Autrement dit, chaque fichier contient le détail de la consommation énergétique d'un seul noeud du réseau.

B- Pour récupérer l'état de l'horloge lors de la transmission et de la réception de paquets, il

faut passer par ces étapes:

1- Accéder à l'application à simuler et la compiler en tapant: **make pc**
2- Taper **export DBG=clock**
Pour afficher des messages en parallèle avec l'horloge, taper **export DBG=clock,usr1**
3- Exécuter **main.exe** en tapant **: /build/pc/main.exe -t=60 -p 10 > NomApp.trace**
4- Accéder au fichier **NomApp.trace**

Il contient des lignes sous la forme suivante :

Moment d'envoi du paquet Heure : Minute : Seconde

2: CLOCK: event handled for mote 2 at **0:0:36.47777400** (347634 ticks).
2: CLOCK: Setting clock interval to 218 @ 0:0:36.47777400
2: j'envoie le paquet de données à la destination 42 ///DBG usr1
.
.
42: j'ai reçu le paquet de données de la source 2 ///DBG usr1

Moment de réception du paquet. Délai de propagation du paquet=36, 60979650-36, 47777400=0, 13202250 secondes

.
.
42: CLOCK: event handled for mote 42 at **0:0:36.60979650** (902286 ticks).
42: CLOCK: Setting clock interval to 231 @ 0:0:36.60979650