Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études
Pour l'obtention du diplôme de Master en Informatique

Option: *Réseaux et systèmes distribués (R.S.D)*

Thème

Evaluation des deux approches des données agrégées sécurisées dans les réseaux de capteurs sans fil : End-To-End et Hop-By-Hop

Réalisé par **:**
- **M^R CHAREF Mohammed Abdelkrim**
- **M^R GHEMBAZA Mohammed Abdelmounaim Chérif.**

Présenté le 11 Mai 2016 devant le jury composé de MM.

Bekara Chakib	**Maître assistant, Université de Tlemcen**	**Président**
Labraoui Nabila	**Maître de Conférences, Université de Tlemcen**	**Encadreur**
Benamar Abdelkrim	**Maître de Conférences, Université de Tlemcen**	**Examinateur**
Benaïssa Mohamed	**Maitre assistant, Université de Tlemcen**	**Examinateur**

Année universitaire: 2015-2016

Remerciements

وَمَـا تَـوْفِـيقِـي إِلَّا بِـاللّه . . .

Nous remercions d'abord la grâce d'Allah, pour nous avoir guidés et éclairés sur la bonne voie du savoir pour continuer ce travail et atteindre les objectifs tracés.

Avant tout, nous tenons à remercier notre encadreur M^me^ **Nabila Labraoui** pour son soutient, son aide, ses conseils précieux et surtout sa patience avec nous durant toutes ces années de formation. Sa disponibilité, ses orientations précises et ses qualités humaines et professionnelles ont contribué, majoritairement, à l'accomplissement de ce travail.

Nous tenons, également, à remercier les membres du jury pour nous avoir honorés en acceptant de juger ce travail.

Nous remercions M^R^ **Bekara Chakib** d'avoir accepté de présider le jury, ainsi que les membres M^R^ **Benamar Abdelkrim** Et M^R^ **Benaïssa Mohamed.**

Nous remercions vivement tous nos enseignants du département d'informatique pour leurs efforts durant cette formation.

Nous n'oublions pas de remercier nos collègues d'études, qui nous ont soutenus et bien intégrés durant ce parcours universitaire.

Nous remercions nos familles, nos amis et tous ceux qui nous ont encouragés à accomplir ce travail.

Dédicace

Je dédie ce modeste travail

A celui qui a consacré sa vie pour nous, qui a assumé son rôle parfaitement et qui l'assume toujours, celui qui a sacrifié son bonheur pour me voir heureux, à mon cher père.

A celle qui a veillé a ma vie et qui le fait toujours, qui m'a entouré de son amour et de sa tendresse et qui ne cesse de m'encourager et me guider, à ma chère mère.

Ce travail vous est dédié et grâce à vous il est accompli.

A mes chers frères, ma chère sœur, mon beau-frère, ma belle-sœur, mes neveux, mes nièces.

A mon grand-père et toute ma grande famille.

A mes amis chacun par son nom et tous ceux qui m'aiment.

A mes collègues de travail qui m'ont encouragé et soutenu durant mes années d'études.

A tous ceux que j'aime, j'apprécie, je respecte et j'honore.

Charef Mohammed Abdelkrim

Dédicace

A

Mes chers parents la cause de ma présence.

Mon épouse qui a fait beaucoup de sacrifice pour moi.

Mes poussins Allaa et Lylia que Dieu me les gardes.

Mes chères retrouvailles Mohamed et Fatima.

Tous mes frères et sœurs.

Mon neveu Habib et mes nièces Nour, Fatima, Youssra et la petite Amina Cécanin.

Celui qui m'a soutenu dans l'ombre Mr Slimani.

Celle que j'admire et que je respecte Elhadja Fatiha.

Tous ceux et celles qui m'ont encouragé de prés ou de loin pour terminer ce manuscrit.

Je dédie ce travail.

Ghembaza Mohamed Abdelmounaim.

Résumé

Le domaine d'application des réseaux de capteurs ne cesse d'accroître avec le besoin d'un mécanisme de sécurité efficace. Le fait que les RCSF traitent des données très souvent sensibles, opérant dans des environnements hostiles et inattendus, la notion de sécurité est considérée comme indispensable. Cependant, à cause de la limitation des ressources et la faible capacité de calcul d'un nœud capteur, ainsi que les ressources limitée en énergie et la consommation énergétique élevée en communication, le choix d'un modèle de déploiement garantissant une sécurité pose de vrais défis.

L'agrégation des données est une technique très utilisée car elle permet de réduire le nombre de message transmis dans le réseau et par conséquent réduire la consommation d'énergie, ainsi améliorer la durée de vie du réseau.

Dans cette mémoire, nous nous sommes intéressés aux problèmes de sécurité liés à l'agrégation des données dans les réseaux de capteurs sans fil. Nous avons étudiés deux approches Bout-en-Bout et Saut-par-Saut et déduire les avantages et les inconvénients de chaque approches.

Mots-clés : RCSF, capteur, agrégation de donnée, sécurité, énergie, Bout-en-Bout, Saut-par-Saut Contiki.

Abstract

The field of application of sensor networks continues to increase with the need for effective security mechanism. The fact that these networks deal with the often sensitive data, operating in hostile and unexpected environments, the concept of security is considered essential. However, because of limited resources and low computing capacity of a sensor node and the limited energy resources and the high energy consumption in communication, choosing a deployment model ensuring security is real challenges.

Aggregation of Data is a widely used technique because it reduces the number of messages transmitted in the network and thereby reduce energy consumption and improve the life of the network.

In this thesis, we focused on safety issues related to data aggregation in wireless sensor networks. We studied two approaches : End-to-End and Hop-by-Hop and deduct the advantages and disadvantages of each approach.

Keywords: WSN, sensor, aggregation of data, security, energy, End-to-End, Hop-by-Hop, Contiki.

ملخص

إن مجال تطبيق شبكات الاستشعار في ازدياد مستمر وكذلك الحاجة إلى آلية فعالة للأمن. لأن هذه الشبكات تتعامل مع بيانات في كثير من الأحيان حساسة، وهي تعمل في بيئات غير محمية، حيث يعتبر مفهوم الأمن ضروري. ومع ذلك، بسبب محدودية الموارد وضعف قدرة الحوسبة وموارد الطاقة المحدودة وارتفاع معدلات استهلاك الطاقة في مجال الاتصالات، فإن اختيار نموذج آمن لنشر شبكات الاستشعار يعتبر تحديات حقيقاً.

إن تقنية تجميع البيانات هو أسلوب يستخدم على نطاق واسع لأنه يقلل من عدد الرسائل المرسلة في الشبكة، وبالتالي الحد من استهلاك الطاقة وإطالة مدة عمل الشبكة.

في هذه الأطروحة، ركزنا على قضايا السلامة المرتبطة بتقنية تجميع البيانات في شبكات الاستشعار اللاسلكية. درسنا طريقتين: النهاية إلى النهاية وقفزة إلى قفزة وبينا مزايا وعيوب كل طريقة.

الكلمات المفتاحية: شبكات أجهزة الاستشعار اللاسلكية، أجهزة الاستشعار، تجميع البيانات ، الأمن، الطاقة، النهاية إلى النهاية، قفزة إلى قفزة، كونتيكي.

Table des matières

Liste des Illustrations

Liste des Figures

Chapitre I : Les Réseaux De Capteurs Sans Fil.

Chapitre II: La sécurité des dans les RCSFs : Taxonomie des menaces et des solutions.

Chapitre III: Implémentation et Discussion des Résultats.

Liste des Tableaux

Introduction Générale

Un réseau de capteurs sans fil (Wireless Sensor Network) est un ensemble de nœuds communiquant via des liens sans fil. Les WSN sont devenus l'une des technologies majeures du 21ème siècle. Grâce à leur mobilité comme des unités embarquées, leur légèreté de composants et leur facilité d'implémentation comme un réseau Ad-Hoc, les WSN sont appliqués de plus en plus dans plusieurs domaines tel que la surveillance environnementale et écologique, l'industrie, l'agriculture, médecine et suivi sanitaire et bien sure en domaine militaire. L'inconvénient majeur d'un WSN est la sécurité puisque ses deux propriétés principales qui sont la mobilité et l'infrastructure sans fil Ad-Hoc impliquent deux problèmes qui sont : la limitation et l'épuisement de ressources énergétiques et les attaques sur les communications sans fil.

Ce travail a comme objectif d'évaluer les performances d'un WSN utilisant l'agrégation sécurisée de données en termes de consommation d'énergie et de temps de délivrance d'information. L'agrégation de données dans un WSN est la division de ce dernier en secteur, où chacun de ces secteurs a un nœud agrégateur (chef) qui effectue une opération précise (médiane, somme, moyenne...etc.) sur les données reçues des capteurs qui lui sont attribués et envoie cette valeur à la station de base. Cette opération permet d'éliminer la transmission redondante des données, ce qui permettra l'économie de l'énergie et la diminution du trafic dans le réseau. En utilisant l'agrégation des données, la sécurité devient un enjeu critique. Un ou plusieurs capteurs compromis peuvent transmettre des données erronées ce qui va affecter directement le résultat d'agrégation des données dans le réseau. D'une manière plus grave, l'agrégateur lui-même peut être compromis, ce qui va fausser les données reçues d'une zone entière.

Dans les WSN, deux approches de sécurisation des données agrégées sont les plus connues, la sécurisation de bout-en-bout (End-to-End) et la sécurisation de saut-par-saut (Hop-by-Hop).

Dans la première approche (End-to-End), Le nœud agrégateur d'une zone effectue l'opération d'agrégation sur des données chiffrées. Il n'a pas besoin de déchiffrer les données

reçues ni de savoir les valeurs réelles captées dans sa zone. Il effectue l'agrégation et transmet le résultat à la station de base qui s'occupe du déchiffrement.
Dans la deuxième approche (Hop-by-Hop), Le nœud agrégateur d'une zone effectue l'opération d'agrégation sur des données en claire, donc il a besoin de déchiffrer les données reçues de chaque capteur pour effectuer l'opération d'agrégation et transmettre le résultat à la station de base.

Nous avons évalué les deux approches selon deux métriques de simulation, à savoir l'énergie consommée et le délai de délivrance d'information.

Le présent document est structuré comme suit :
Chapitre 1 : Les Réseaux De Capteurs Sans Fil.
Chapitre 2 : La sécurité des dans les RCSFs : Taxonomie des menaces et des solutions.
Chapitre 3 : Implémentation et Discussion des Résultats.
Et nous avons terminé par une conclusion.

Chapitre I. Les Réseaux De Capteurs Sans Fil (WSN: Wireless Sensor Networks)

I. Introduction

Les avancées technologiques récentes confortent la présence de l'informatique et de l'électronique au cœur du monde réel. De plus en plus d'objets se voient ainsi équipés de processeurs et de moyens de communication mobiles, leur permettant de traiter des informations mais également de les transmettre. Les réseaux de capteurs sans-fil entrent dans ce cadre. En effet, ceux-ci sont constitués d'un ensemble de petits appareils, ou capteurs, possédant des ressources particulièrement limitées, mais qui leur permettent néanmoins d'acquérir des données sur leur environnement immédiat, de les traiter et de les communiquer.

1. Les Réseaux de capteurs sans fil

1.1 Qu'est-ce qu'un nœud capteur ?

Un nœud capteur sans fil est un petit dispositif électronique doté d'un ou plusieurs capteurs qui sont capables de mesurer une valeur physique environnementale (température, lumière, pression, etc.) ou physiologique (glycémie, tension, etc.), et de la communiquer à un centre de contrôle via une station de base [1].

Un capteur est capable de transformer une grandeur physique observée (température, humidité, etc.) en une grandeur utilisable (intensité électrique, position d'un flotteur). Il possède au moins un transducteur dont le rôle est de convertir une grandeur physique en une autre tel qu'ADC [1].

Par ailleurs, grâce aux avancées technologiques, principalement dans le domaine de la miniaturisation, les capteurs sont devenus des éléments de très petite taille. Ils sont ainsi dotés de moyens leur permettant : de stocker les résultats de leurs observations, d'effectuer un certain nombre de traitements sur ces résultats et de les transmettre au monde réel via une communication sans fil [1].

1.1.1 Le capteur intelligent

Les capteurs intelligents "Smart Sensors" sont des dispositifs matériels dans lesquels coexistent le(s) capteur(s) et les circuits de traitement et de communication. Leurs relations

avec des couches de traitement supérieures vont bien au-delà d'une simple "transduction de signal"[2].

Les capteurs intelligents sont des "capteurs d'information" et non pas simplement des capteurs et des circuits de traitement du signal juxtaposés. De plus, ces capteurs ne sont pas des dispositifs banalisés car chacun de leurs constituants a été conçu dans l'objectif d'une application bien spécifique [2].

La Figure I.1 illustre un exemple de capteur intelligent.

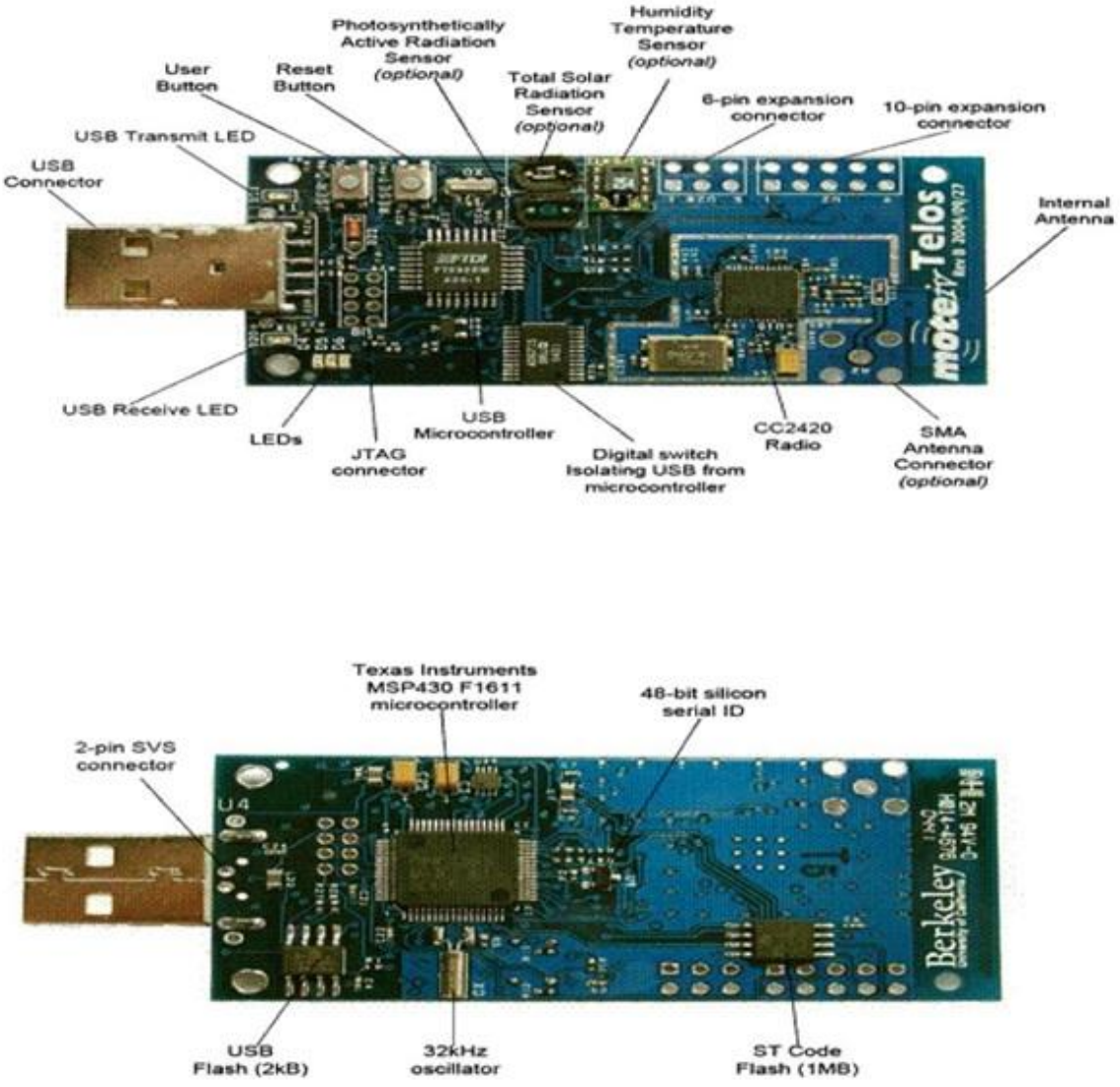


Figure I.1: Anatomie d'un capteur TelosB [03]

1.1.2 Les composants d'un nœud capteur

Les capteurs sans fil sont conçus comme de véritables systèmes embarqués, dotés de moyens de traitement et de communication de l'information, en plus de leur fonction initiale de relever des mesures. Ils représentent une révolution technologique des instruments de mesure, issue de la convergence des systèmes électroniques miniaturisés et des systèmes de communication sans fil [4].

La Figure I.2 présente une anatomie générale d'un capteur sans fil.

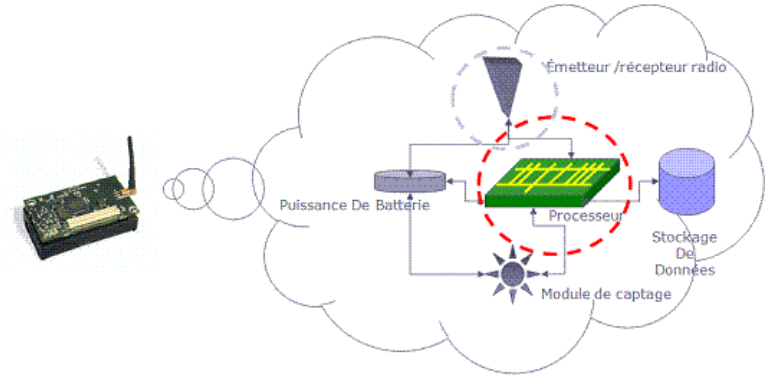


Figure I.2: Les composants d'un nœud capteur [3]

Selon le type d'application, il existe une variété de capteurs qui peuvent être regroupés en trois classes : Les capteurs optiques, les capteurs thermiques et les capteurs mécaniques. Une architecture matérielle applicable à la plupart des capteurs intelligents est proposée sur la figure suivante [5].

Un capteur contient quatre unités de base : l'unité de captage, l'unité de traitement, l'unité de transmission, et l'unité de contrôle d'énergie. Selon le domaine d'application, des modules supplémentaires peuvent être ajoutés tel qu'un système de localisation (GPS), ou bien un système générateur d'énergie (cellule solaire). Quelques micro-capteurs plus volumineux, sont dotés d'un système mobilisateur chargé de les déplacer en cas de nécessité[5].

Néanmoins, un capteur sans fil est composé fondamentalement de quatre unités élémentaires :

1. **Unité d'acquisition de données** : Ce composant est l'unité qui contient le ou les capteurs embarqués sur le nœud. Habituellement, un convertisseur analogique-numérique (CAN) convertit les signaux provenant des capteurs (signaux analogiques) en signaux interprétables par l'Unité de Traitement (signaux numériques) [4].

2. **Unité de traitement des données** : Elle est généralement constituée d'un microcontrôleur dédié et de la mémoire. Les microcontrôleurs utilisés dans le cadre de réseaux de capteurs sont à faible consommation d'énergie. Leurs fréquences sont assez faibles, moins de 10 MHz pour une consommation de l'ordre de 1 mW. Une autre caractéristique est la taille de leur mémoire qui est de l'ordre de 10 Ko de RAM pour les données et de 10 Ko de ROM pour les programmes (Holger, et al., 2005). Cette mémoire consomme la majeure partie de l'énergie allouée au microcontrôleur, c'est pourquoi on lui adjoint souvent de la mémoire flash moins coûteuse en énergie. Outre le traitement des données, le microcontrôleur commande également toutes les autres unités notamment le système de transmission [4].

3. **Unité de transmission de données** : Elle est le plus souvent constituée d'un transmetteur radio qui fournit au capteur la capacité de communiquer avec les autres au sein d'un réseau. Les composants utilisés pour réaliser la transmission sont des composants classiques. Ainsi on retrouve les mêmes problèmes que dans tous les réseaux sans fil : la quantité d'énergie nécessaire à la transmission augmente avec la distance. Pour les réseaux sans fil classiques (LAN, GSM) la consommation d'énergie est de l'ordre de plusieurs centaines de milliwatts, et on se repose sur une infrastructure alors que pour les réseaux de capteurs, le système de transmission consomme environ 20 mW et possède une portée de quelques dizaines de mètres. Certaines technologies radio permettent de changer la fréquence et la puissance de transmission [4].

4. **Unité de puissance** : Comme il est souhaitable de s'affranchir de toute connexion par câble, le capteur doit disposer de sa propre source d'énergie qui est responsable de répartir l'énergie disponible aux autres modules et de réduire les dépenses en mettant en veille les composants inactifs par exemple. Cette unité se trouve généralement sous la forme de batteries standards de basse tension. A titre indicatif, ce sera souvent une pile AA normale d'environ 2.2-2.5 *Ah* fonctionnant à 1.5 V [4].

En fonction des applications pour lesquelles ils sont conçus, les capteurs sans fil pourraient également avoir d'autres modules, comme *une unité de localisation*, afin d'identifier leur position géographique, par exemple en utilisant un récepteur GPS ou une technique de triangulation. Certaines applications pourraient aussi avoir besoin de capteurs équipés d'*un mobilisateur* pour qu'ils puissent se déplacer [4].

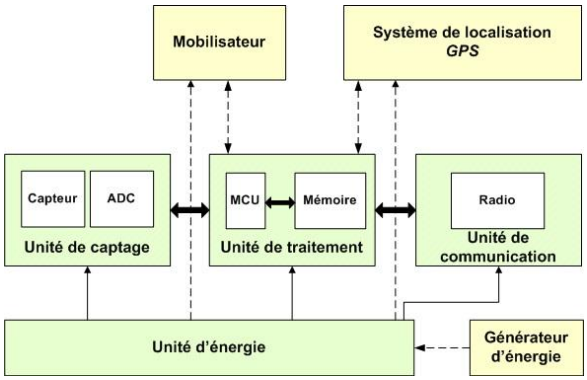


Figure I.3: Architecture d'un nœud capteur [5]

1.2. Les réseaux de capteurs sans fil

1.2.1. Définition des réseaux de capteurs sans fil

Un RCSF est constitué généralement d'un grand nombre de nœuds capteurs car ces derniers sont sujets à des pannes accidentelles ou intentionnelles. Ces nœuds communiquent entre eux selon une certaine topologie de réseau afin d'acheminer les informations à un centre de contrôle distant de la zone de leur déploiement. La mise en place d'un RCSF est soumise à plusieurs contraintes parmi lesquelles la conservation de l'énergie des nœuds capteurs, la fidélité de monitoring et la fidélité de routage. Néanmoins, la conservation de l'énergie et la tolérance aux pannes restent parmi les grands défis des RCSF. Dans cette optique plusieurs contributions ont été proposées dans la littérature. Ces contributions visent à minimiser la consommation d'énergie afin d'optimiser l'autonomie des nœuds qui constituent le réseau par suite garantir une longue durée de vie de réseau et assurer la fidélité de routage c'est-à-dire garantir l'acheminement de l'information d'une station vers une autre station à n'importe quel moment au cours du fonctionnement du réseau. La Figure I.4 résume un fonctionnement simple d'un RCSF [1].

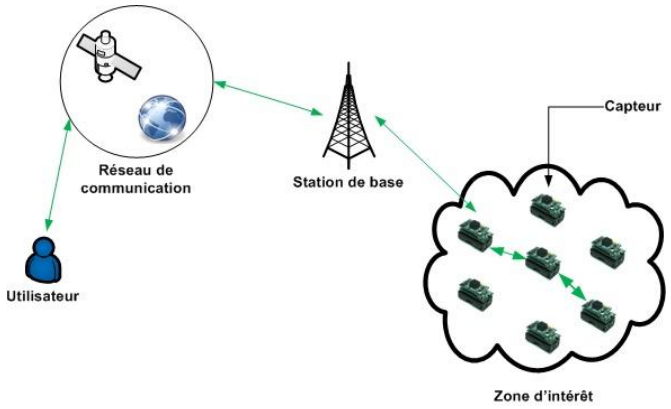


Figure I.4: Exemple d'un réseau de capteurs [5].

1.2.2. Architecture des réseaux de capteur sans fil

Pour mieux comprendre les systèmes physiques et par la suite, les différentes stratégies adoptées pour dimensionner et architecturer un RCSF, nous recourrons à des modèles aussi simples que possible. Dans cette section, nous définissons plusieurs modèles qui sont utilisés dans les RCSF. Ainsi, nous utilisons les modèles de nœuds, les modèles de communication, les modèles de détection ou d'acquisition, les modèles de consommation d'énergie [6].

1.2.2.1 Modèle de nœud

Selon l'application et la structure choisie, un RCSF peut contenir différents types de nœuds [6].

- **Un nœud régulier** est un nœud doté d'une unité de transmission et d'une unité de traitement de données. L'unité de transmission de données est responsable de toutes les émissions et réceptions de données via un support de communication sans fil pouvant être de type optique (comme dans les nœuds Smart Dust) ou de type radiofréquence (comme dans les nœuds Stargate). L'unité de traitement de données est composée d'une mémoire, d'un microcontrôleur et d'un système d'exploitation spécifique. Elle est responsable du traitement des données en provenance ou au départ de l'unité de transmission. Ces deux unités sont alimentées par une batterie embarquée. Selon le domaine d'application, un nœud peut être équipé d'unités supplémentaires ou optionnelles comme un système de localisation (Global Positioning System ou GPS, etc.) pour déterminer sa position, ou bien un système générateur d'énergie (cellule photovoltaïque, etc.), ou encore un système mobile pour lui permettre de changer sa position ou sa configuration en cas de nécessité.

- **Un nœud capteur** ou nœud source est un nœud régulier équipé d'une unité d'acquisition ou de détection. L'unité d'acquisition est généralement dotée d'un capteur ou plusieurs capteurs qui obtiennent des mesures analogiques (physiques et physiologiques) et d'un convertisseur Analogique/Numérique qui convertit l'information relevée en un signal numérique compréhensible par l'unité de traitement.

- Un **nœud actionneur ou robot** est un nœud régulier doté d'une unité lui permettant d'exécuter certaines tâches spécifiques comme des tâches mécaniques (se déplacer, combattre un incendie, piloter un automate, etc.)

- Un **nœud puits** est un nœud régulier doté d'un convertisseur série connecté à une seconde unité de communication (GPRS, Wi-Fi, WiMax, etc.). La seconde unité de communication fournit une retransmission transparente des données provenant de nœuds capteurs à un utilisateur final ou d'autres réseaux comme internet.

- Un **nœud passerelle (ou gateway)** est un nœud régulier permettant de relayer le trafic dans le réseau sur le même canal de communication.[06]

Pour optimiser certains paramètres comme la durée de vie du réseau ou le délai de livraison des données, certains travaux se sont focalisés sur l'architecture (plat, hiérarchique, multi-niveaux) des RCSF. Ces architectures définissent le plus souvent les rôles joués par les nœuds dans un RCSF. Nous distinguons principalement 3 rôles à savoir:[6]

- **Nœud Source (NS)**: dont le rôle principal est de détecter les phénomènes physiques ou physiologiques se produisant dans son environnement immédiat afin de les transmettre, directement ou via multiples sauts, à un utilisateur final. C'est en fait un nœud capteur.

- **Nœud Relais (NR)**: ils ont pour rôle d'agréger et de retransmettre les mesures provenant des NS afin que celles-ci parviennent à un utilisateur final. Dans une architecture à plat, quelques travaux considèrent généralement un NS comme un NR.

Dans une architecture à 2 niveaux, un nœud passerelle joue le rôle de NR pour un ou plusieurs nœuds sources. Dans une telle configuration réseau, la capacité de transmission du NR est supposée généralement plus grande que celle du NS.

- **Nœud Collecteur (NC)** de données : ils ont pour rôle de collecter les mesures provenant des nœuds sources et éventuellement de les agréger. Généralement, un "Cluster-Head " ou chef de cluster est utilisé comme NC dans une architecture hiérarchique où les NS sont partitionnés en plusieurs groupes.

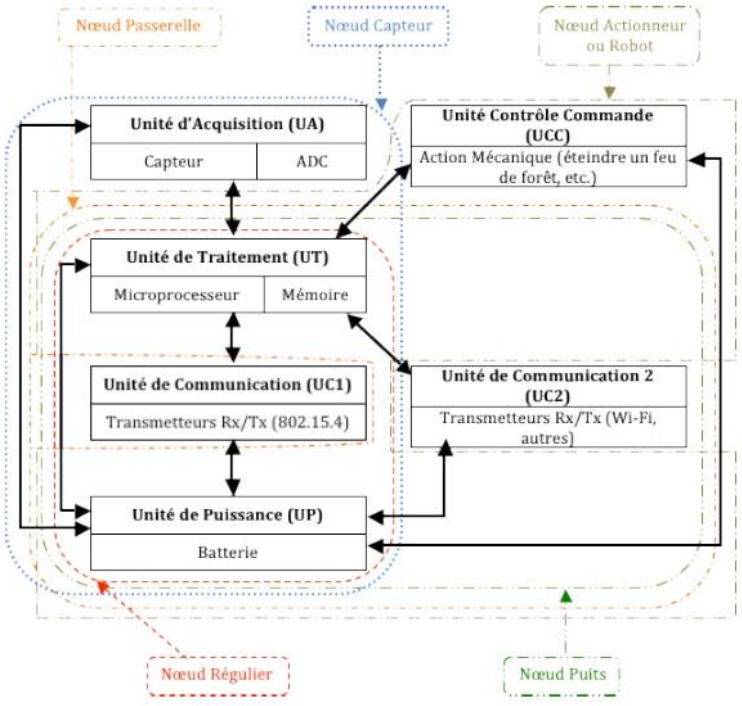


Figure I.5: Architecture des différents types de nœuds : régulier, capteur, robot, puits, passerelle [06]

1.2.2.2 Modèle de la topologie

Un RCSF est composé d'un ensemble de nœuds capteurs qui s'occupent de collecter les données des capteurs et de les transmettre à l'utilisateur via l'internet ou par satellite. L'acheminement des données dans les RCSF se fait selon plusieurs schémas de routage qui sont sous-jacents à des topologies. Il existe plusieurs topologies pour les réseaux de capteurs:[1]

- **Topologie en étoile :** la topologie en étoile est un système uni-saut c'est-à-dire les nœuds terminaux communiquent directement avec le nœud coordinateur (station de base). Cette topologie est simple et elle demande une faible consommation d'énergie, mais la vulnérabilité de la station base pourrait bloquer tout le réseau. En outre, cette topologie exige que la distance entre les nœuds et la station de base soit limitée [1].

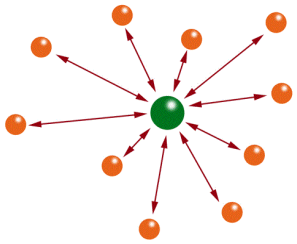


Figure I.6: Une architecture typique de réseau en étoile

- **Topologie en toile :** (Le réseau maillé ou mesh) la topologie en toile est un système multi-sauts. La communication entre les nœuds et la station de base pourrait être directe ou via un schéma de routage multi-sauts. Chaque nœud a plusieurs chemins pour envoyer des données. Cette topologie permet le passage à l'échelle et la tolérance aux pannes, mais elle demande une consommation d'énergie plus importante [1].

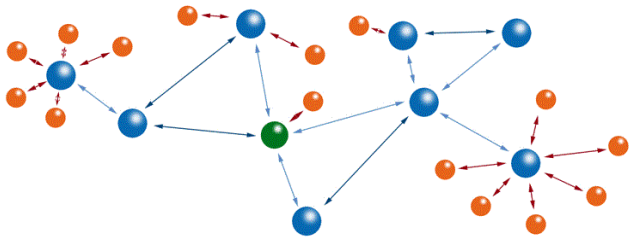


Figure I.7: Une architecture typique de réseau maillé

- **Topologie hybride :** la topologie hybride est une combinaison des deux topologies (étoile et toile). Les stations de base forment une topologie en toile et les nœuds autour d'elles sont en topologie étoile. Elle assure la minimisation d'énergie dans les réseaux de capteurs [1].

1.2.2.3. Modèle de collecte d'information

Il y a deux méthodes pour collecter les informations d'un réseau de capteurs.

- Collecte d'information à la demande :

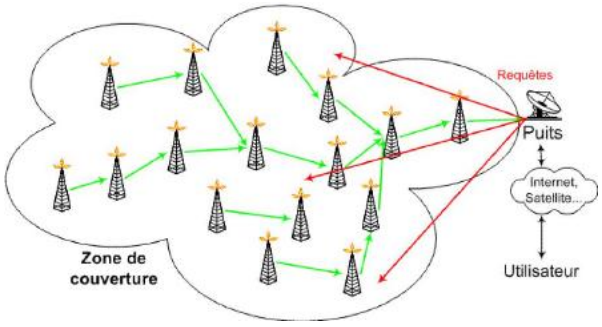


Figure I.8 : Collecter les informations à la demande. [3]

Lorsque l'on souhaite avoir l'état de la zone de couverture à un moment T, le puits émet des brodcasts vers toute la zone pour que les capteurs remontent leur dernier relevé vers le puits. Les informations sont alors acheminées par le biais d'une communication multi-sauts[3].

- Suite à un événement

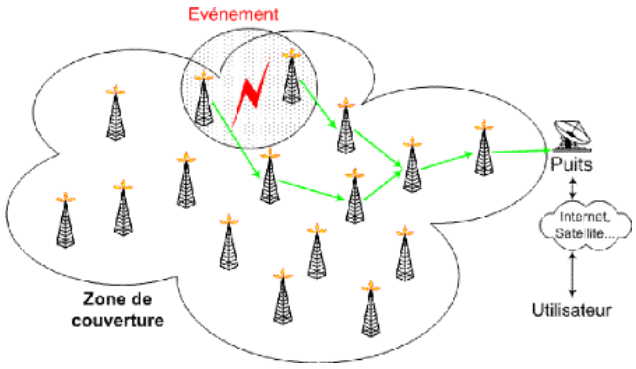


Figure I.9: Collecter les informations Suite à un événement. [3]

Un événement se produit en un point de la zone de couverture (changement brusque de température, mouvement...), les capteurs situés à proximité remontent alors les informations relevées et les acheminent jusqu'au puits [3].

1.3. Domaines d'application

Les capteurs ouvrent de nouveaux horizons à la gestion de l'information. Ils forment une "peau virtuelle" avec le monde réel qui nous informe des événements physiques se déroulant autour de nous. Cela a naturellement attiré plusieurs domaines d'applications qui ont mis en œuvre les facilités que les capteurs leur offrent. En voici une liste non exhaustive [4].

- **Le domaine militaire** : comme pour beaucoup d'autres domaines, les applications militaires ont été les locomotives de la recherche pour les réseaux de capteurs. Pour les militaires, un réseau de capteurs offre des avantages très précieux. Il s'agit d'un réseau qui s'installe rapidement, dynamiquement et sans aucune infrastructure. Ainsi, il offre un atout de taille pour surveiller les mouvements de l'ennemi, communiquer à bas coût entre les unités avec une logistique peu compliquée[4].
- **La surveillance environnementale** : la petite taille et les capacités relativement grandes au niveau de calcul et de communication des capteurs permettent de les placer aux endroits que les humains ne peuvent ou ne veulent pas accéder, comme par exemple les grandes forêts, les volcans, les profondeurs des océans, les régions polaires, ou encore d'autres planètes que la terre [4].
- **L'industrie** : les industriels s'intéressent au potentiel des capteurs pour diminuer les coûts du contrôle et de la maintenance des produits, de la gestion de l'inventaire, de la télésurveillance après vente, etc.. [4].
- **Les domaines urbains et domotique** : les capteurs entrent de plus en plus dans nos vies quotidiennes. Dans le milieu urbain, les capteurs sont déjà utilisés pour la localisation des bus, pour des tickets électroniques et pour la sécurité. Une des applications est la surveillance du trafic routier avec les réseaux de capteurs déployés sur les autoroutes. De plus, les maisons, les bâtiments, les bureaux équipés de capteurs intelligents permettent de construire des systèmes pervasifs où l'information est omniprésente [4].
- **Le domaine médical** : la recherche sur l'usage des capteurs intelligents dans le domaine médical inclut les moyens d'hospitalisation à domicile, l'intégration des micro-capteurs "dans" le corps (e.g. construire un BAN - Body Area Network) et la gestion des urgences. Parmi les applications les plus utiles, on cite la télésurveillance des signes

vitaux et des niveaux d'activité à domicile des personnes âgées ou handicapées ainsi que le contrôle à distance des données physiologiques [4].

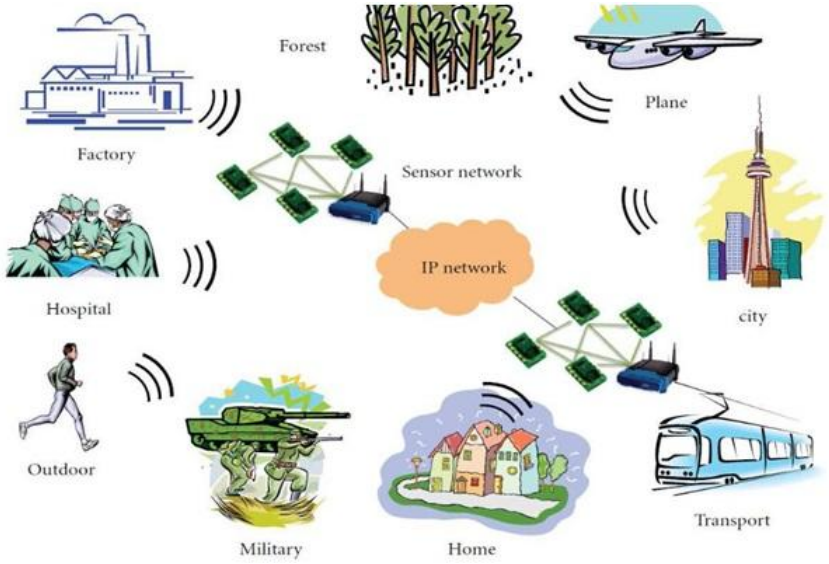


Figure I.10: Quelques domaines d'application des RCSF [7].

1.4 Facteurs influençant l'architecture des WSN

Les principaux facteurs et contraintes influençant l'architecture des réseaux de capteurs peuvent être résumés comme suit :

- **La tolérance aux pannes :** Certains nœuds peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, un problème physique ou une interférence. Ces anomalies ne doivent pas affecter le reste du réseau. Cette contrainte mesure la capacité de maintenir les fonctionnalités du réseau sans interruption dues à une panne intervenue sur un ou plusieurs capteurs [1].
- **Passage à l'échelle :** La plupart des protocoles conçus pour RCSF ont été établis pour des réseaux dont le nombre de nœuds est moyen. Cependant, les performances de ces protocoles risquent de se dégrader lorsque le nombre de nœuds augmente. D'où on dit qu'un protocole s'adapte au passage à l'échelle si et seulement si quand le nombre de nœuds augmente dans le réseau, les performances de ce dernier ne doivent pas se dégrader [1].
- **Le coût de production :** Souvent, les réseaux de capteurs sont composés d'un très grand nombre de nœuds. Le prix d'un est critique afin de pouvoir construire un réseau de surveillance traditionnel [1].

- **Environnement :** Les nœuds capteurs doivent être conçus d'une manière à résister aux sévères conditions de l'environnement : forte chaleur, pluie, humidité…[1]
- **Les medias de transmission :** 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 normé. On utilise le plus souvent l'infrarouge, le Bluetooth et les communications radio Zigbee [1].
- **La consommation d'énergie :** Les capteurs sont généralement alimentés par des batteries et dans certaines applications ils sont déployés dans des zones hostiles où l'intervention humaine est presque impossible pour toute opération de maintenance. De ce fait, le remplacement de la batterie est impossible. Cette énergie est consommée par les différentes unités du capteur afin de réaliser leurs tâches, c'est pour cette raison que les recherches actuelles se concentrent principalement sur les moyens de réduire la consommation pour garantir une longue durée de vie aux réseaux déployés [1].
- **Les contraintes matérielles :** La principale contrainte matérielle est la taille du capteur. La petite taille des capteurs ne permet de le doter par d'autres unités telles qu'un grand nombre de batteries et un mobilisateur [1].
- **La topologie de réseau :** Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phrases :

  - Déploiement.
  - Post-déploiement.
  - Redéploiement de nœuds additionnels.[1]

1.5. Principaux domaines de recherche

Dans la littérature, plusieurs travaux de recherche visent à proposer des solutions optimales et efficaces à un ou plusieurs problèmes des RCSFs illustres précédemment. Les principaux domaines de recherche abordes dans les RCSFs sont les suivants :

1. *Efficacité énergétique* : en raison de la ressource énergétique limitée, plusieurs solutions, à la fois du matériel et des logiciels, ont été proposées afin d'optimiser l'utilisation de l'énergie [8].

2. *Localisation* : vu le très grand nombre des nœuds capteurs sur un RCSF et leur déploiement d'une manière ad hoc, de nombreux systèmes de coordonnées spatiales et

virtuelles ont été proposés, auxquels les nœuds capteurs peuvent s'identifier pour se localiser dans le RCSF.

3. *Routage* : plusieurs protocoles de routage ont été proposés pour les RCSFs pour minimiser les couts de communication, afin de réduire la consommation énergétique.

4. *Sécurité* : les applications utilisant les RCSFs ont souvent besoin d'un niveau de sécurité élevé. Or, de part leurs caractéristiques, la sécurisation des RCSFs est à la source de beaucoup de travaux scientifiques et techniques proposant des solutions de sécurité efficaces[9].

2. Agrégation de données dans un RCSF

Dans un capteur, la problématique principale concerne la consommation d'énergie: en effet, ces derniers doivent restés opérationnels le plus longtemps possibles, dans des conditions parfois difficiles (ex : lâchés par avion sur les parois d'un volcan). Comme il n'est pas possible de recharger leur énergie ni changer les piles par exemple (on ne sait pas toujours où se trouvent les capteurs), il est nécessaire d'économiser au maximum l'énergie consommée par ces derniers [4].

On estime que la transmission des données d'un capteur représente environ 70% de sa consommation d'énergie.

De plus, les réseaux de capteurs étant assez denses en général, cela signifie que des nœuds assez proches en terme de distance peuvent capter les mêmes données (température, pression, humidité équivalentes par exemple) et donc il apparaît nécessaire d'introduire une approche intéressante qui est d'agréger les données basées sur le principe que la station de base n'a pas besoin de toutes les données collectées par chaque capteur en raison de leur redondance dans le réseau mais seulement d'un agrégat de données effectué au niveau d'un nœud appelé agrégateur en utilisant des simples fonctions d'agrégat telles que :la moyenne, la médiane, la somme, le max ou le min…qui permettent à partir d'une série de n messages reçus par un « chef de zone » (capteur chef d'une zone) de ne renvoyer vers le puits qu'un seul message résumant l'information contenue dans ces n messages (voir Figure I.13) et qui aide à préserver l'énergie en évitant la duplication de l'information et donc d'augmenter la durée de vie du réseau [4, 10].

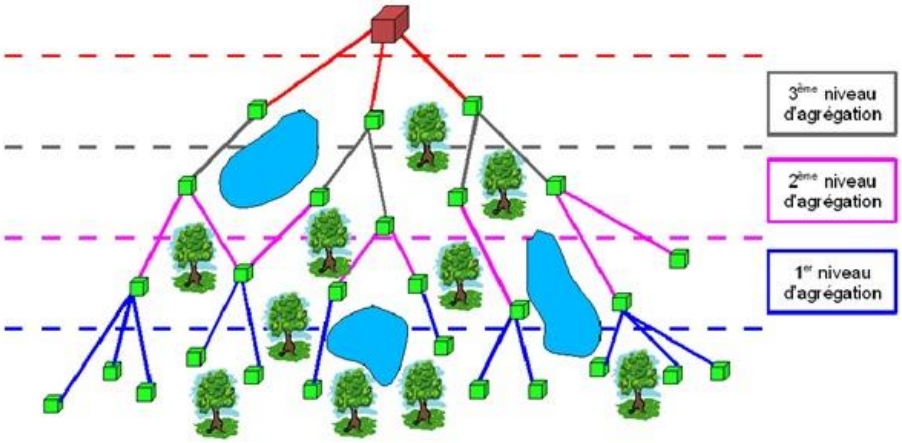


Figure I.11: Arbre d'agrégation de données [10].

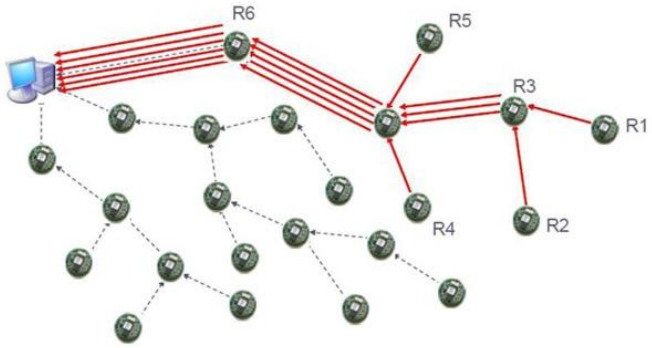


Figure I.12: Exemple sans agrégation [4].

Au total, 18 messages sont envoyés sur le réseau de capteurs. En utilisant le mécanisme d'agrégation de données, on obtient un total de 7 messages envoyés sur le réseau :

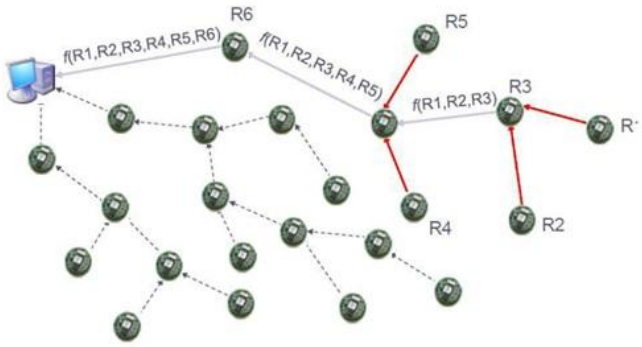


Figure I.13: Exemple avec agrégation de données [4].

2.1. Les techniques d'agrégation

Les techniques d'agrégation peuvent former deux types de structures. Les techniques centralisées s'appuient sur la construction des grappes d'agrégation, et les techniques distribuées construisent des arbres d'agrégation.

2.1.1. Agrégation dans les grappes

Dans ce cas, le réseau est organise en groupes de grappes de nœuds. Chaque grappe contient une tête de grappe qui représente le nœud agrégateur (Figure 14). Il existe plusieurs protocoles d'agrégation qui se basent sur la structure en grappe. Parmi eux, le protocole LEACH (*Low-Energy Adaptive Clustering Hierarchy*) pour l'économie d'énergie. Ce protocole s'exécute en deux phases. Durant la première phase, une tête de grappe est élue et les grappes sont ainsi formées et organisées. Ensuite, chaque tête de grappe collecte les informations de tous les nœuds de sa grappe.

Pendant la deuxième phase, toutes les informations collectées seront agrégées et envoyées au puits. Afin de prolonger la durée de vie de la tête de grappe, ainsi que celle du réseau, les nœuds prennent le relai de tête de grappe à tour de rôle [12].

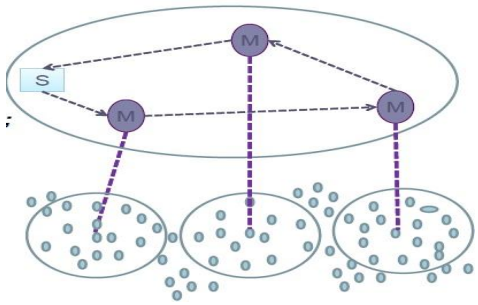


Figure I.14: Agrégation dans les grappes [12].

2.1.2. Agrégation dans les arbres

La structure du réseau est un arbre dont la racine est le puits et les nœuds intermédiaires sont les agrégateurs (Figure 15). Chaque nœud intermédiaire reçoit les informations de ses fils, les agrège et les envoie a son père. Un exemple d'application, les réseaux déployés pour mesurer la température moyenne d'une zone géographique. Chaque

nœud intermédiaire additionne les données reçues de ces fils, ajoute la valeur de sa mesure, calcule la moyenne et envoie le résultat a son père [12].

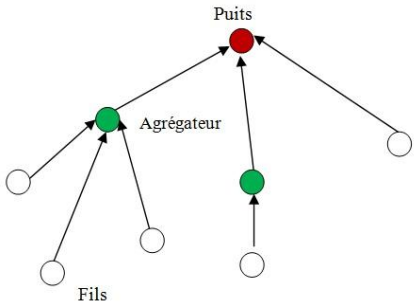


Figure I.15 : Arbre des nœuds pour l'agrégation [12].

Conclusion

Les progrès réalisés ces dernières décennies dans les domaines de la microélectronique, de la micromécanique, et des technologies de communication sans fil, ont permis de produire à un coût raisonnable des composants de quelques millimètres cubes de volume. De ce fait, un nouveau domaine de recherche s'est créé pour offrir des solutions économiquement intéressantes et facilement déployables à la surveillance à distance et au traitement des données dans les environnements complexes et distribués : les réseaux de capteurs sans fil.

Dans ce chapitre, nous avons donné une description générale des réseaux de capteurs sans fil ainsi qu'une description de l'approche d'agrégation. Dans le chapitre suivant, nous allons abordé la problématique de sécurité des données agrégées.

Chapitre II. La sécurité des dans les RCSFs : Taxonomie des menaces et des solutions

I. Introduction

Les réseaux de capteurs sans fil sont constitués de nœuds déployés en grand nombre en vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome. Ces réseaux ont un intérêt particulier pour les applications militaires, environnementales, domotiques, médicales, et bien sûr les applications liées à la surveillance des infrastructures critiques. Ces applications ont souvent besoin d'un niveau de sécurité élevé. Or, de part de leurs caractéristiques (absence d'infrastructure, contrainte d'énergie, topologie dynamique, nombre important de capteurs, sécurité physique limitée, capacité réduite des nœuds,...), la sécurisation des réseaux de capteurs est à la source, aujourd'hui, de beaucoup de défis scientifiques et techniques [4].

1. La sécurité dans les WSN

Comme nous l'avons déjà mentionné dans la section précédente, les réseaux de capteur sans fil ont un intérêt particulier pour les applications militaires, environnementales, domotiques, médicales, et bien sûr les applications liées à la surveillance des infrastructures critiques. La conception de ces applications suppose que tous les nœuds engagés sont coopératifs et dignes de confiance. Cependant, ceci n'est pas le cas dans les déploiements du monde réel, où les nœuds sont exposés à différentes types d'attaques qui peuvent carrément endommagé le bon fonctionnement du réseau. Ces attaques exploitent essentiellement l'incertitude du canal de communication et le déploiement aléatoire des nœuds capteurs dans des zones difficiles à surveiller. Garantir la sécurité de ce type de réseau est une tâche difficile, surtout quand les nœuds sont constitués d'engins électroniques peu onéreux avec des capacités matérielles limitées. Le cas échéant, utiliser des protections physiques est, dans beaucoup de situations, quasiment impraticable. Capturer des nœuds est alors une possibilité intéressante pour les attaquants [4]. Néanmoins, les WSN ne peuvent compter sur l'intervention humaine pour faire face aux tentatives d'un attaquant pour compromettre le réseau ou gêner ses propres opérations [4].

Dans cette section, nous présentons un aperçu sur les problèmes de sécurité dans les réseaux de capteurs sans fil. Premièrement, nous présentons les défis de sécurité pour les WSN qui rendent la sécurité pour ce type de réseaux assez dure. Ensuite, nous présentons une taxonomie des attaques et discutons les besoins de sécurité requis par les protocoles.

1.1. Propriétés à impact majeur sur la sécurité

La sécurité des WSN peut être classifiée en deux grandes catégories (1) la sécurité opérationnelle et (2) la sécurité de l'information. L'objectif de la sécurité relative à l'opération est d'assurer la continuité de fonctionnement du réseau en entier même si une partie de ses composants a été attaquée (service de disponibilité). Quant à la sécurité relative à l'information, son objectif est que la confidentialité de l'information ne doit jamais être divulguée et que l'intégrité et l'authentification de l'information doivent toujours être assurées. Alors qu'il peut sembler que la sécurité de l'information peut aisément être réalisée avec la cryptographie, ils existent néanmoins trois obstacles qui rendent l'achèvement des objectifs cités ci-dessus non trivial dans les réseaux de capteurs sans fil : les ressources très limitées, la communication sans fil et le couplage étroit avec l'environnement [4].

***1.1.1. Limitation de ressources*:** l'énergie est peut-être la contrainte la plus forte aux capacités d'un nœud capteur. La réserve d'énergie de chaque nœud doit être conservée pour prolonger sa durée de vie et ainsi que celle de l'ensemble du réseau. Dans la plupart du temps, l'information transmise est redondante vus que les capteurs sont généralement géographiquement co-localisés. La plupart de cette énergie peut donc être économisée par agrégation de données. Cela exige une attention particulière à détecter l'injection de fausses données ou la modification défectueuse de données, lors des opérations d'agrégation au niveau des nœuds intermédiaires.[13].

1.1.2. La communication sans fils multi-sauts: en plus de fournir un déploiement simple, la communication sans fil a l'avantage d'offrir l'accès à des endroits difficilement accessibles tels que des terrains désastreux et hostiles. Malheureusement, la portée de la communication radio des "motes" est limitée en raison de considérations énergétiques. La communication multisauts est donc indispensable pour la diffusion des données dans un RCSF. Cela introduit de nombreuses failles de sécurité à deux niveaux différents: attaque de la construction et maintenance des routes, et attaque des données utiles par injection, la modification ou la suppression de paquets. En outre, la communication sans fil introduit d'autres vulnérabilités à la couche liaison en ouvrant la porte à des attaques de brouillage et de style déni de service par épuisement des batteries[13].

1.1.3. Couplage étroit avec l'environnement: la plupart des applications de RCSF exigent un déploiement étroit des nœuds à l'intérieur ou à proximité des phénomènes à

surveiller. Cette proximité physique avec l'environnement conduit à de fréquentes compromissions intentionnelles ou accidentelles des nœuds. Comme le succès des applications RCSF dépend également de leur faible coût, les nœuds ne peuvent pas se permettre une protection physique inviolable. Par conséquent, un adversaire "bien équipé" peut extraire des informations cryptographiques des nœuds capteurs. Comme la mission d'un RCSF est généralement sans surveillance, le potentiel d'attaquer les nœuds et de récupérer leur contenu est important. Ainsi, les clefs cryptographiques et informations sensibles devraient être gérées d'une manière qui augmente la résistance à la capture des nœuds [13].

La figure suivante résume les problèmes de sécurité émergeant des caractéristiques d'un RCSF et les solutions à entreprendre :

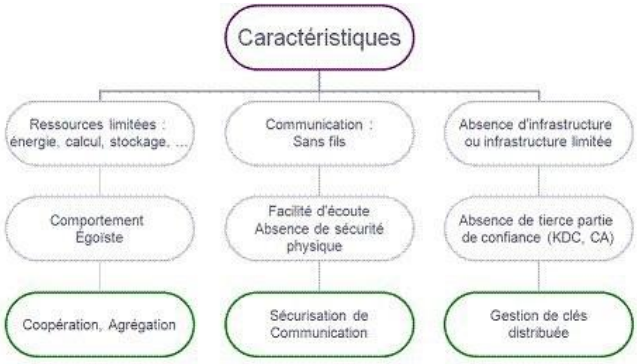


Figure II.1: Sécurité dans les RCSF : propriétés, challenges et solutions [13].

1.2. Les besoins de sécurité typiques aux WSN

Pour déterminer des objectifs de sécurité, il faudra connaitre ce qu'on doit protéger. Les réseaux de capteurs partagent certaines caractéristiques des réseaux mobiles ad hoc mais aussi possèdent des propriétés spécifiques aux WSN, discutées dans la section précédente. Donc les objectifs de sécurité englobent ceux des réseaux traditionnels et les objectifs issus des contraintes intrinsèques aux WSNs. Parmi les principaux objectifs de sécurité, nous citons[04] :

1.2.1. L'authentification

Elle permet de coopérer au sein des WSN sans risque, en contrôlant et en identifiant les participants. Elle apparaît comme la pierre angulaire d'un réseau de capteur sans fil sécurisé. En effet, on ne peut assurer une confidentialité et une intégrité des messages

échangés si, dès le départ, on n'est pas sûr de communiquer avec le bon nœud. Si l'authentification est mal gérée, un attaquant peut se joindre au réseau et injecter des messages erronés.

L'utilisation de Code d'Authentification de Message (CAM), ou MAC en anglais (Message Authentication Code), permet d'assurer à la fois l'authentification de l'origine et l'intégrité du message [4].

1.2.2. La confidentialité

Une fois les parties authentifiées, la confidentialité reste un point important, étant donné la communication sans fil des WSN. Elle consiste à préserver le secret des messages échangés et ne pas les révéler aux adversaires. La confidentialité peut être assurée par l'usage de la cryptographie à clé symétrique ou asymétrique [4].

1.2.3. L'intégrité

Elle assure que les données reçues n'ont pas été altérées durant leur transit dans le réseau de manière volontaire ou accidentelle. Elle peut être assurée par l'utilisation des fonctions de hachage cryptographiques qui permettent d'obtenir pour chaque message une empreinte numérique.

Les fonctions MD2 (Message Digest 2), MD5, SHA-1 (Secure Hash Algorithm 1) sont des exemples de quelques fonctions de hash les plus utilisées [4].

1.2.4. La disponibilité

Elle signifie que le réseau est disponible pour assurer ses services et autoriser les parties communicantes lorsque ceci est nécessaire. Cette propriété reste difficile à assurer dans les WSN étant donné les contraintes qui pèsent sur ces réseaux, à savoir : topologie dynamique, ressources limitées des nœuds de transit, communications sans fil pouvant être facilement brouillées ou perturbées [4].

1.2.5. La fraîcheur

Ce dernier service permet de garantir que les données échangées sur le réseau sont actuelles et ne sont pas une réinjection de précédents échanges interceptés par un attaquant [4]

1.3. Les types de vulnérabilités des WSN

Les vulnérabilités sont les faiblesses d'un réseau que l'attaquant exploite afin de gagner des privilèges. Il y a deux types de vulnérabilités dans un réseau de capteurs WSN [14]:

1.3.1. La vulnérabilité physique est un moyen d'attaque, qui permet à l'attaquant de changer en partie un capteur, en modifiant par exemple son code de programmation, ou en copiant les clés de protection afin de les réutiliser dans une nouvelle attaque. Un réseau de capteurs est vulnérable aussi aux modifications de son environnement, où un attaquant peut modifier les valeurs d'un capteur local, lui permettant ainsi d'avoir un accès aux commandes de contrôle du réseau WSN [14].

1.3.2. La vulnérabilité logique réside dans les programmes et les protocoles. Elle se présente sous quatre formes :

- **Les défauts de conception** permettent l'utilisation d'un protocole qui viole le mode d'utilisation, tout en se conformant à la spécification du protocole. Par exemple, un manque d'authentification dans un protocole de gestion de puissance peut permettre de mettre n'importe quel capteur en sommeil à plusieurs reprises.
- **Les défauts d'implémentation** sont des erreurs dans la construction du matériel ou dans le codage du logiciel. Par exemple, une erreur de dépassement de mémoire, peut entraîner une violation d'accès et une mise en panne.
- **Les défauts de configuration** sont le résultat de défauts de paramétrages pour un attaquant.
- **L'épuisement des ressources** est possible même si la conception, l'implémentation, et la configuration sont correctes. Un attaquant générant de grandes quantités de trafic peut inonder un des liens réseau de la victime. Une mauvaise authentification de l'allocation de mémoire ou de l'exécution de code peut également permettre à un attaquant de consommer les ressources du capteur subissant l'attaque, et de causer un déni de service [14].

1.4. Les attaques contre les WSN

Les différentes spécificités des réseaux de capteurs sans fil citées précédemment (énergie limitée, faible puissance de calcul, utilisation des ondes radio, etc..) exposent les

réseaux de capteurs à de nombreuses menaces. Si certaines de ces menaces peuvent se retrouver dans les réseaux ad-hoc, d'autres sont spécifiques à ce type de réseau et s'attaquent tout particulièrement à l'énergie limitée des capteurs.

Dans les cas d'attaques que l'on retrouve dans les réseaux de capteurs sans fil, un attaquant peut chercher à récupérer les informations du réseau en écoutant le médium, si le réseau n'encrypte pas ses données. Dans ce cas de figure, on parlera d'attaque passive, l'attaquant ne cherchant ici qu'à écouter et récupérer les informations [15].

Les attaques dans le RCSF connaissent plusieurs classifications, mais les plus connues sont regroupées selon les catégories ci-dessous [16] :

1.4.1. Classification selon l'origine

-**Attaque interne:** Elle se produit à l'intérieur du réseau. Dans ce cas, l'intrus est aperçu par les autres nœuds comme étant un nœud normal. Ce phénomène ce produit lorsque le noud malveillant connait la clé de chiffrement et peut enclencher le processus de cryptage et décryptage. Par conséquent, il peut accéder aux messages chiffrés échangés entre les nœuds. Cette menace est la plus sévère et la plus difficile à détecter.

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

1.4.2. Classification selon la nature

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

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

1.4.3. Classification selon la couche ciblée

-**Les attaques ciblant la couche physique:** L'attaque *Jamming* est la plus fréquente dans la couche physique d'un RCSF. Celle-ci vise à créer des interférences pour occuper les canaux et empêcher les capteurs de communiquer normalement.

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

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

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

1.5. Bloques fonctionnels de la sécurité dans les RCSF.

Comme illustré à la figure suivante, on distingue quatre blocks fonctionnels des solutions de sécurité dans les RCSF :

- la gestion de clés,
- la sécurité du routage,
- la sécurité de l'agrégation de données,
- et la sécurité de l'accès au canal [13].

Nous allons détailler dans la suite la sécurité de l'agrégation de données.

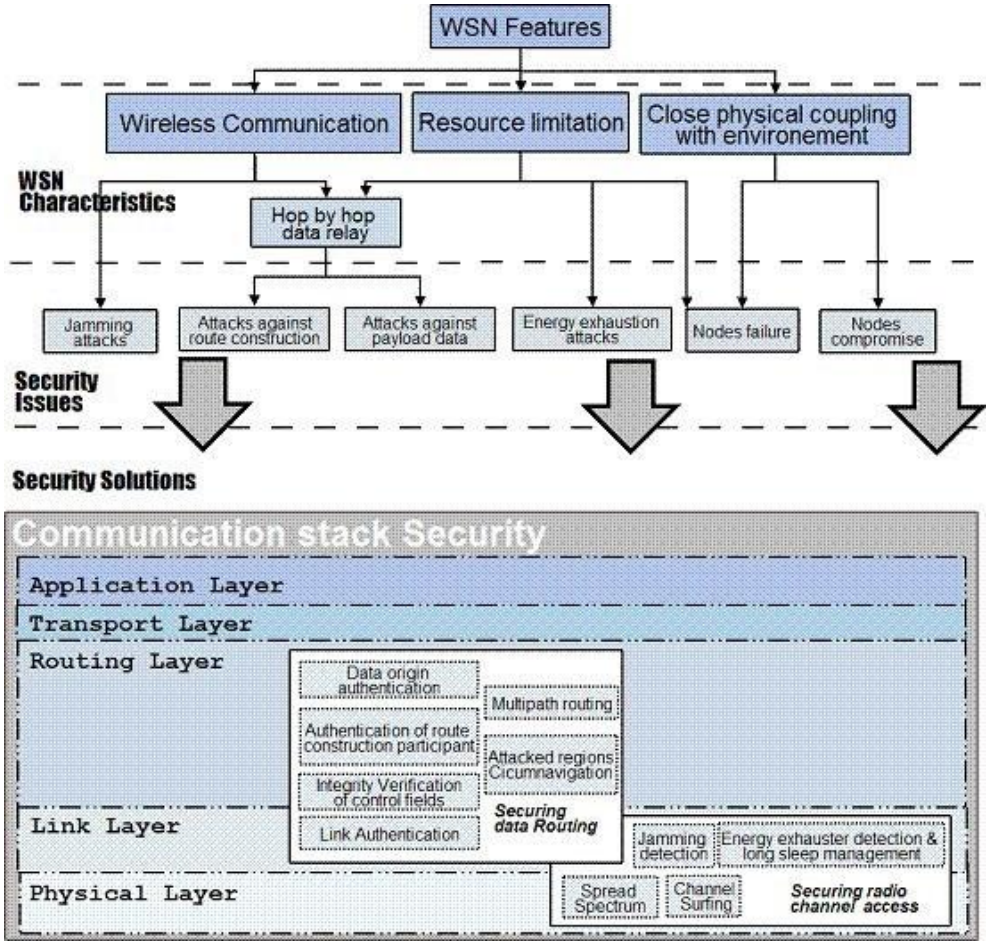


Figure II.2: Taxonomie des challenges et solutions de sécurité dans les RCSF [13].

2. Sécurité de l'agrégation dans les RCSF.

2.1. Attaques sur l'agrégation de données dans les RCSF

L'agrégation de données est nécessaire dans les RCSF pour minimiser les transmissions redondantes et donc économiser de l'énergie. Pour réaliser une opération d'agrégation, un nœud intermédiaire doit avoir accès aux données transmises par ses paires pour calculer l'information utile en utilisant une fonction d'agrégation comme : la moyenne, le maximum, le minimum etc.

Un nœud malicieux peut alors attaquer ce schéma en injectant de fausses données dans le réseau ou en falsifiant le résultat d'une opération d'agrégation. Dans ce cas, le nœud malicieux réussira à falsifier l'information captée dans toute une zone.

La figure suivante montre le risque qui peut être encouru par une mauvaise opération d'agrégation.

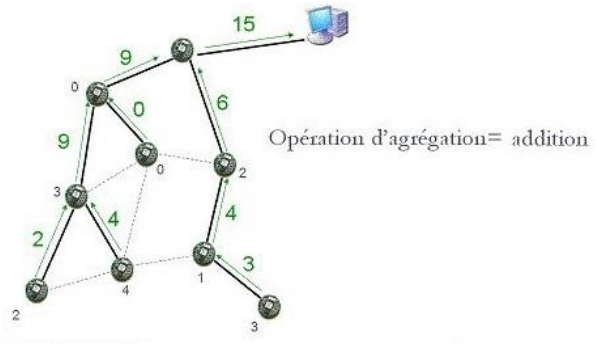


Figure II.3: Fonctionnement correcte de l'agrégation.[13].

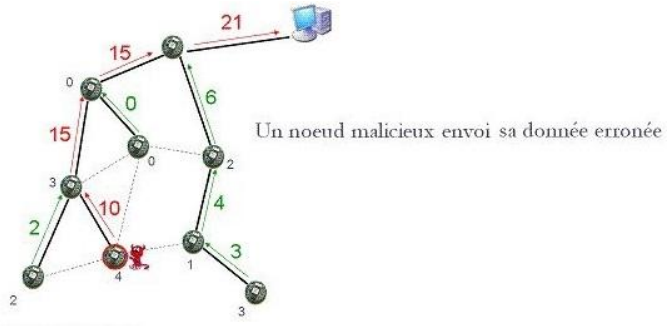


Figure II.4: Cas d'un malicieux nœud qui injecte une fausse donnée[13].

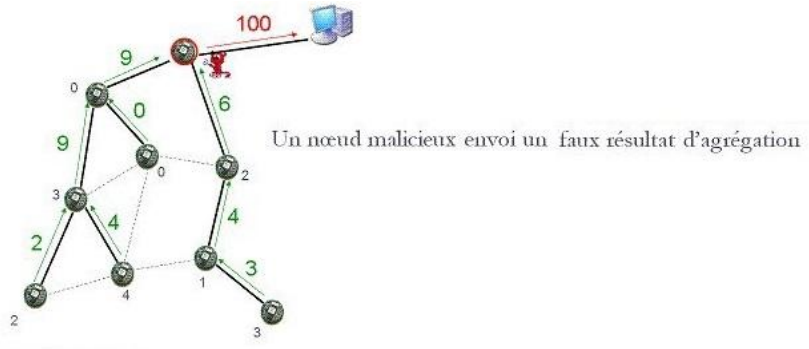


Figure II.5: Un malicieux falsifie le résultat d'une agrégation. [13].

Toute la difficulté est comment permettre aux nœuds de relais d'avoir accès aux résultats intermédiaires pour calculer l'information utile par agrégation, et rendre cette opération exempt de risque de falsification, suppression ou modification ?

2.2. Sécurité de l'agrégation de données dans les RCSF.

Nous classifions les solutions de sécurité de l'agrégation dans les RCSF en deux grandes catégories: les approches distribuées où l'agrégation se fait dans un arbre de routage recouvrant les nœuds capteurs, et les approches centralisées où l'agrégation se fait dans des clusters. Nous raffinons davantage cette classification en sous-catégories selon les pairs qui établissent des associations de sécurité pour protéger l'agrégation[17] : l'approche à base d'associations de sécurités saut-par-saut, l'approche basée sur des associations de sécurité de bout en bout. Dans la première approche, la vérification de l'intégrité des données se fait par les nœuds capteurs eux-mêmes avec l'aide de la station de base (SDAP [18], SAWN [19], SecureDAV [20], RSDA [21]). Dans la seconde approche, c'est la station de base seule qui effectue tout le processus de vérification de l'intégrité des données (CMT [22], ASAP [23]).

2.2.1. Solutions basées sur le cryptage de bout en bout.

Dans cette catégorie on utilise des mécanismes cryptographiques qui sécurisent l'information captée de bout en bout tout en permettant aux nœuds intermédiaires de réaliser les opérations d'agrégation. Dans cette catégorie, la vérification de l'information ne se fait généralement qu'au niveau du collecteur, ce qui engendre une forte contamination de la fausse information [13].

Les protocoles de cette catégorie utilisent une clé partagée entre chaque nœud et le nœud collecteur pour garantir l'intégrité des données transmises dans le réseau. Comme les contenus des données sont cryptés, les nœuds utilisent un type de cryptographie particulier appelé "Privacy Homomorphism (PH) pour pouvoir exécuter l'agrégation.

Un algorithme est PH si et seulement si en ayant E(x) et E(y) on peut calculer E(x¤y) sans décrypter x et y. Ainsi il vérifie la propriété suivante :

$E_{K1}(x_1)$ ¤ $E_{K2}(x_2)=E_{K1+K2}(x_1$ ¤ $x_2)$, où K_i sont les clés et x_i sont les données.

Le point unique de vérification dans ce type de protocoles est le nœud collecteur. Ce dernier ayant toutes les clés utilisées pour crypter les données dans le réseau.

Le protocole CMT [22] proposé par Castelluccia, Myldetun et Tsudik est basé sur l'hypothèse que chaque nœud utilise une clé symétrique partagée entre ce nœud et le nœud

collecteur. L'idée de ce protocole est que chaque nœud fait l'addition modulaire entre sa clé stockée et sa donnée. Pendant la phase d'acheminement des données, l'agrégation se fait sur ces données qui sont déjà cryptées. L'algorithme suivant montre les différentes étapes de ce protocole [13]:

Algorithme de CMT[13]	
Paramètre :	
	Sélection d'un grand nombre entier M.
Cryptage :	
	Le message $m \in [0, M-1]$. Aléatoirement générer une clé $k \in [0, M-1]$. $C = (m + k) mod M$.
Décryptage :	
	$m = (c - k) mod\ M$.
Agrégation :	
	$c_{12} = (c_1 + c_2)\ mod\ M$.

Algorithme II.1: Algorithme CMT

La taille du paquet dans ce protocole dépend de la taille de M, et une seule addition modulaire suffit pour l'agrégation et le cryptage. Ainsi, ce protocole ne consomme pas beaucoup d'énergie [13].

2.2.2. Solutions basées sur le cryptage de proche en proche.

Dans ce cas, la véracité de l'information est vérifiée de proche en proche et son rejet peut se faire à n'importe quel niveau de l'arbre couvrant le RCSF.

Si on prend par exemple le protocole SAWN [19] (Secure Aggregation for Wireless Networks) issus de cette famille il suppose que deux nœuds consécutifs ne peuvent pas être compromis simultanément.

Il se base sur la vérification à deux sauts : un nœud vérifie si l'agrégation des données de ses petits fils, réalisée par son fils, est correcte.

La vérification de l'agrégation se fait d'une manière différée dans le temps en utilisant le protocole µTESLA pour l'authentification des clés utilisées dans l'authentification des données et de leurs agrégations.

Dans ce qui suit nous supposons que les nœuds ont le moyen de vérifier l'authenticité des clés partagées entre les nœuds et la SB, lorsque cette dernière révèle les clés pour la vérification.

Chaque nœud feuille transmet sa lecture à son père. Les messages incluent la lecture des données du nœud, son id, ainsi qu'un MAC calculé grâce à la clé K_{Ai}. Cette dernière est partagée entre le nœud A et la station de base, mais n'est pas encore connue par les autres capteurs. Le nœud père stocke le message ainsi que son MAC jusqu'à la révélation de clé K_{Ai} par la station de base. A cet instant, il vérifiera le MAC et envoi une alarme en cas de différence. L'agrégation des lectures est exécutée dans chaque étape intermédiaire. Les nœuds attendent pendant un temps indiqué pour recevoir des messages de leurs fils et retransmettent ensuite les messages et les MACs qu'ils reçoivent directement de leurs fils immédiats. Les nœuds agrègent les données qu'ils reçoivent de leurs petits-fils (via leurs fils) et transmettent le MAC de la valeur d'agrégation. Après l'arrivée de tous les messages à la station de base, cette dernière révèle les clés temporaires des nœuds. Une fois que la clé (K_{Ai}) est révélée, les nœuds passent à la clé temporaire suivante (K_{Ai+1}).

Pour mieux comprendre Considérons l'arbre d'agrégation illustré par la figure II.6. L'exemple illustre le i-ème tour où l'en utilise les clés K_{xi} pour authentifier les messages transmis:

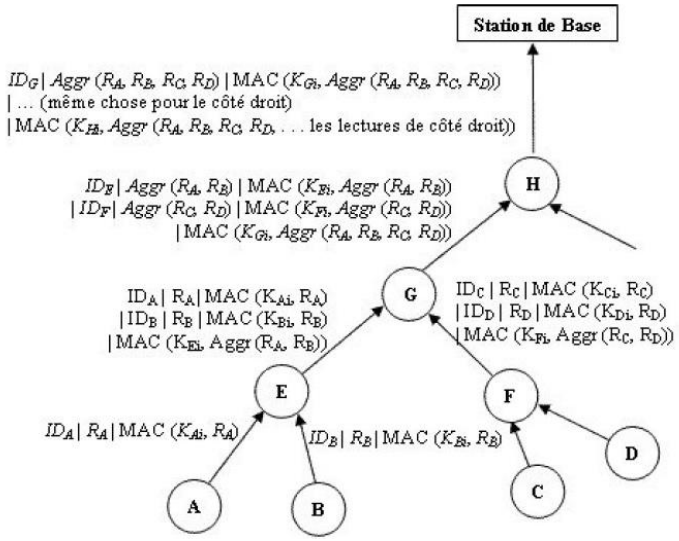


Figure II.6: Exemple d'arbre d'agrégation sécurisé avec SAWN [13].

La transmission se fait comme suite : Les nœuds A, B, C et D envoient des données à la station de base via l'arbre d'agrégation construit avec un protocole de routage.

- Les nœuds feuilles envoient des données à leur père. Les messages incluent des MACs calculés avec la clé d'authentification courante:

$$A ==> E: R_A \mid ID_A \mid MAC(K_{Ai}, R_A)$$

Équation 1

Chaque clé est utilisée pour authentifier un seul message ce qui empêchera l'attaque par rejoue.

- Les nœuds intermédiaires reçoivent les messages de leurs fils. Le nœud père ne peut pas encore vérifier le MAC car la clé du fils ne sera révélée que pendant la phase de vérification. Pour le moment, le père stocke le message et le MAC. Le nœud intermédiaire attend les paquets des fils, et envoie ensuite un message à son père contenant les lectures des fils, leurs MACs, ainsi que le MAC calculé sur la valeur d'agrégation:

$$E => G : R_A \mid ID_A \mid MAC(K_{Ai}, R_A) \mid R_B \mid ID_B \mid MAC(K_{Bi}, R_B) \mid MAC\big(K_{Ei}, Aggr(R_A, R_B)\big)$$

Équation 2 : Le message reçue par le nœud père.

Il n'y a aucun besoin de transmettre la valeur d'agrégation calculée, puisque G peut calculer $Aggr(R_A,R_B)$ depuis les valeurs R_A et R_B. Ce n'est pas aussi nécessaire de transmettre l'ID_E à G, parce que G connaît la topologie du réseau donc peut déterminer le nœud qui envoie le message.

- Le nœud G reçoit les messages des nœuds E et F. Pour chacun d'eux, G calcule les valeurs de l'agrégation des lectures de ses petits-fils c'est-à-dire (A, B, C et D). Il transmet alors les valeurs agrégées de ses petits-fils, l'ID de ses fils et leurs valeurs de MAC. G calcule aussi et transmet le MAC de la valeur d'agrégation suivante :

$$Aggr(R_A, R_B, R_C, R_D) = Aggr\big(Aggr(R_A, R_B), Aggr(R_C, R_D)\big)$$

Équation 3 : La fonction d'agrégation.

Puisque la fonction de l'agrégation est connue à tous les nœuds, le MAC calculé par E authentifiera la valeur calculée par G. Les lectures des capteurs et les valeurs de MAC reçues à partir de E et F sont stockées pour vérification postérieure.

$$G ==> H: ID_E \mid Aggr(R_A, R_B) \mid MAC\big(K_{Ei}, Aggr(R_A, R_B)\big) \mid ID_F \mid Aggr(R_C, R_D)$$

$$\left|MAC\left(K_{Fi}, Aggr(R_C, R_D)\right)\right| MAC\left(K_{GI}, Aggr(R_A, R_B, R_C, R_D)\right)$$

Équation 4 : Stockage des Données.

De la même façon, le nœud H reçoit des messages de G et d'une autre branche, et transmet à son tour le message agrégé à la station de base.

Noter que la longueur du message n'augmente pas si le réseau était plus profond.

- La station de base reçoit le message de H. Elle peut calculer la valeur de l'agrégation finale, Aggr(R_A, R_B, R_C, R_D, ...) en utilisant Aggr(R_A,R_B,R_C,R_D) et les autres valeurs de ses nœuds fils [13].

Le but du protocole SAWN [19] est d'authentifier toutes les lectures qui ont participé à la valeur d'agrégation, sans pour autant recevoir toutes ces lectures. Pour valider les données (lecture des nœuds et valeurs d'agrégation), la station de base révèle la clé courante K_{xi} qu'elle partage avec chaque nœud x du réseau, en émettant un seul message contenant toutes ces clés. Ce message de révélation des clés est authentifié par un MAC en utilisant une clé authentifiée grâce à µTESLA. Si un nœud détecte un message erroné dans l'étape de validation de données, il envoie un message d'alarme. Une alarme est émise par un parent quand il détecte que le MAC d'agrégation d'un fils est contradictoire avec les données des petits-fils, ou bien quand les MAC des données eux-mêmes sont erronés [13].

Conclusion

Dans ce chapitre nous nous somme intéressés à la sécurité des réseaux de capteurs sans fil on a cité les principales attaques et défaillances des RCSF sans oublier de citer les différents technique de sécurité en projetant la lumière sur les étapes de chiffrement et déchiffrement de la collecte de l'information par le nœud collecteur jusqu'à sa réception par la station de base chose qu'on va tenter d'étudier plus profondément dans la partie qui suit en simulant le fonctionnement des RCSFs avec l'approche End-to-End et l'approche Hop-by-Hop.

Chapitre III. Implémentation et Discussion des Résultats.

I. Introduction

Dans cette partie on va premièrement présenter les plateformes logicielles utilisées ainsi que la configuration matérielle, ensuite on va présenter les différents scénarios développés puis on va passer à la présentation des résultats obtenus pour finaliser cette partie en argumentant les graphes et résultats.

1. Environnement de développement.

Notre simulation a été réalisée dans l'environnement logiciel suivant :

- Système d'exploitation installé dans le micro: Microsoft Windows 8.
- Lecteur des machines virtuelles : VMPlayer "7.1.0 build-2496824".
- Système d'exploitation installé sous la machine virtuelle : Ubuntu 14.04.4 LTS.
- Un Système d'exploitation pour capteur : Contiki "Instant Contiki 2.7"
- Un Simulateur de réseaux de capteur sans fil : Cooja.
- Le simulateur MATLAB R2010a pour tracer les graphes.

1.1. Contiki

1.1.1. Présentation.

Contiki[28] est un système open source, léger, flexible et générique qui s'appuie sur un modèle de fonctionnement hybride[24].Ce système a été développé par un groupe de développeurs de l'industrie et du monde universitaire par Adam Dunkels de l'institut suédois d'informatique en 2002. Destiné à être embarqué dans des capteurs miniatures ne disposant généralement que de ressources limitées, Contiki a présenté l'idée d'utiliser la communication IP dans des réseaux de capteurs basse consommation. En plus il supporte les protocoles IPV6 et 6LOWPAN cela s'avère particulièrement utile dans la mesure où les nœuds communiquent en IPV6 et utilisent le standard 802.15.4 définie par l'IEEE.

Contiki contient deux piles de communications : uIP et Rime[25].

uIP est une petite pile de TCP/IP RFC-CONFORME qui permet a Contiki de communiquer sur internet.

Rime est une pile de communication légère conçue pour des radios basse puissance. Il fournit une vaste gamme de communications primitives.

La programmation d'application dans ce système d'exploitation se fait dans le langage C et pour permettre la concurrence, contiki utilise des protothreads qui sont en fait des threads légers sans pile spécialement conçus pour les environnements avec peu de mémoire comme les réseaux de capteurs [25].

1.1.2. Architecture de Système Contiki.

Pour économiser de la mémoire, l'approche basée sur les événements a été, dans un premier temps, privilégiée. Les incertitudes sur la taille de la pile d'exécution et le nombre de processus à prévoir sont ainsi évitées. Cependant, les opérations longues telles que la cryptographie de données s'accordent mal avec l'utilisation des événements par la monopolisation du système pour une durée conséquente. Pour cette raison, dans un second temps, le système Contiki s'est vu ajouter un composant qui lui permet de fonctionner comme un système multitâche. Ce composant est une bibliothèque de fonctions optionnelle appelée explicitement par le programme qui en a besoin. Cette bibliothèque permet la gestion des processus et de leur pile d'exécution respective[11].

Le cœur du système Contiki est composé de différents éléments :

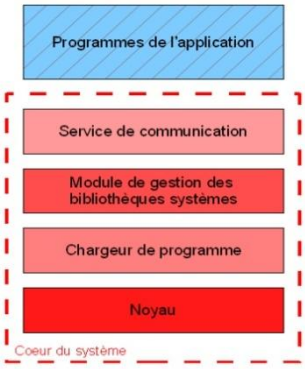


Figure III.1: Architecture du Système Contiki[24].

- **le noyau**

Le noyau gère des événements asynchrones et synchrones. Les premiers sont placés dans une file d'attente après leur appel. Les traitements associés à ce type d'événement sont donc déclenchés après un certain délai. A l'inverse, la réponse à un événement synchrone est quasi immédiate[11].

Dans le système Contiki, des fonctions utilisées par les traitements associés à différents événements sont regroupés dans un même service. Cette notion est très proche de celle d'une bibliothèque partagée. Les bibliothèques comme les services sont modifiables et remplaçables dynamiquement en cours d'exécution. Ils offrent donc des possibilités de reconfiguration très intéressantes et utiles si l'on considère le cadre d'utilisation des RCSF[11].

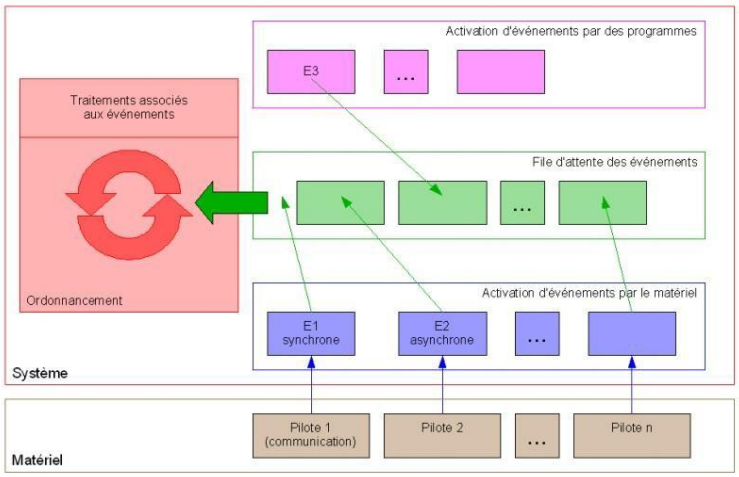


Figure III.2 : Gestion des événements par le systeme Contiki[11].

- **Le chargeur de programme**

L'ajout de programmes configure le noyau pour une application donnée. Le système ne contient pas de niveau d'abstraction pour l'économie d'énergie mais offre des informations comme la taille de la file d'attente des événements pour que l'application puisse réaliser cette opération[11].

- **Le module de gestion des bibliothèques systèmes**

En ce qui concerne les bibliothèques, selon leur emplacement, leur reconfiguration est plus ou moins envisageable. Celles qui font partie du cœur du système, le plus souvent intégrées au module de gestion des bibliothèques, sont considérées comme statiques. Les bibliothèques associées aux programmes de l'application et les services ont vocation à être remplacés dynamiquement[11].

- **La pile de communication avec les pilotes pour le matériel**

Le service de communication établit le lien entre l'application et le matériel à l'aide des pilotes associés. Le noyau du système intervient dans la gestion des événements

synchrones activés durant les phases de réception et d'émission d'un paquet ou d'un message. De manière générale, la communication entre services est obtenue par publication d'événements[11].

Jusqu'à présent, seul le fonctionnement basé sur les événements a été abordé. Logiquement, par rapport au système TinyOS, certains principes sont en commun, d'autres différents. Comme nous l'avons indiqué précédemment, le mode multitâche est assuré par une bibliothèque spécifique[11].

Sous le système Contiki, pour simplifier la transformation d'une application en un ensemble d'événements, un nouveau principe dénommé « protoprocessus» («protothreads») a été développé et est également disponible grâce à une bibliothèque[26].

Les «protoprocessus» sont semblables dans leur fonctionnement aux événements à la différence près, qu'à la manière des processus, ils peuvent être bloqués. Cependant, ce blocage est localisé à des emplacements précis symbolisés par la présence de la primitive PT_WAIT_UNTIL (cond1). Le déblocage se produit quand la condition « cond1 » est vérifiée. Comme pour les événements, les «protoprocessus» partagent la même pile d'exécution[11].

Contiki ne permet pas l'exécution d'applications en temps réel. Bien qu'il soit possible de charger la bibliothèque permettant d'exécuter des threads en parallèle, le multi-threading est chargé par-dessus l'ordonnanceur événementiel. Une tâche lancée par l'ordonnanceur en mode événementiel ne peut pas être interrompue par les tâches exécutée en mode multi-thread.

Comme nous avons pu le constater tout au long de cette présentation dédiée au système Contiki, celui-ci offre une architecture hybride novatrice mais qui s'appuie surtout sur un système basé sur les événements. On passe d'un mode de fonctionnement à l'autre sans pouvoir réellement les combiner sauf si l'on utilise les «protoprocessus».

1.1.3. Les caractéristiques.

Un système d'exploitation pour capteur en réseau a différentes caractéristiques :

- **Empreinte mémoire**

L'espace mémoire utilisé par le système d'exploitation et par l'application doit être suffisamment faible pour être contenu dans la mémoire du capteur. Une configuration typique de Contiki (le noyau et le chargeur de programmes) consomme 2 kilooctets de RAM et 40 kilooctets de ROM[25].

- **Consommation électrique :**

L'énergie électrique, souvent apportée par une batterie de piles, peut être problématique à renouveler. Si des systèmes de captage, comme des éléments photovoltaïques, éoliens, ou autres peuvent être utilisés dans certains cas, les recherches scientifiques explorent les possibilités de réduire la consommation des capteurs. L'élément le plus consommateur est le module radio[25].

La réduction de temps de transmission et de réception radio est primordiale. Pour cela, le module radio est activé lorsque nécessaire, et arrêté ou mis en veille le reste du temps. Mais lorsque le module radio est arrêté, le capteur ne reçoit pas les messages qui lui sont destinés. Un réveil périodique risque d'être inutile, et donc de consommer de l'énergie de façon inefficace. Pour gérer cette problématique, Contiki propose par défaut ContikiMAC, un mécanisme conçu pour rester en communication avec le réseau efficacement, tout en permettant la mise hors tension du module radio 99 % du temps. D'autres techniques permettent de limiter la consommation telle que le compactage des données à transférer, le pré-calcul (afin de ne transmettre que les données réellement utiles), mais aussi une optimisation du routage. Dans certains cas, il peut être utile de stocker des informations dans une base de données locale au capteur, en effet, si le capteur doit envoyer en ensemble de mesures semblables à des résultats déjà envoyés à un autre moment, il peut être préférable d'envoyer une référence à ces données déjà envoyées (Parcourir 100 enregistrements dans une base coûte moins d'énergie que de transmettre un paquet radio)[25].

- Communications :

Contiki implémente deux mécanismes de communication[25]:

1. La couche de protocole Rime :

Rime est une légère couche de communication qui réduit la complexité d'uIP. Elle fournit à la couche applicative un jeu d'instructions de communication, permettant les différentes connexions avec les capteurs voisins. Les protocoles de la pile Rime sont disposés dans un mode en couches, où les protocoles les plus complexes sont implémentés en utilisant les protocoles moins complexes. Toutes les communications en Rimes ont identifiées par une chaine de 16 bits.

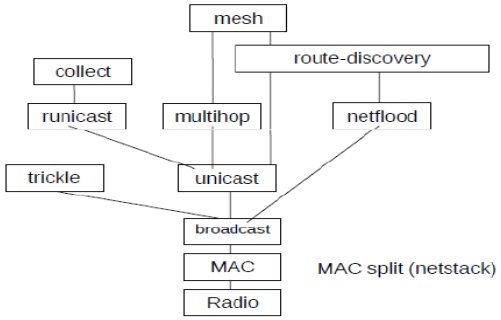


Figure III.3 : Carte Générale de la pile rime[27].

2. La couche uIP :

Elle supporte les protocoles IP, TCP, UDP et ICMP. Contiki implémente uIPv4 et uIPv6 et ce dernier est la première implémentation d'IPv6 pour capteurs miniatures. Pour les communications radio via le protocole IEEE 802.15.4, Contiki implémente 6LoWPAN. Lors de communications radio suivant la norme IEEE 802.15.4 communément utilisée par les capteurs, la taille d'un paquet est limitée à 127 octets, ce qui est insuffisant pour transmettre un paquet IPv6 dont la taille maximum est de 1 280 octets. L'IETF a créé une couche d'adaptation 6LoWPAN qui se situe entre les couches liaison et réseau du modèle OSI. 6LoWPAN permet la compression de l'entête du paquet IPv6 ainsi que la fragmentation et le réassemblage des datagrammes. Pour le routage d'IPv6 à travers le réseau de capteur, Contiki intègre ContikiRPL dans la couche uIPv6, une implémentation du protocole de routage RPL.

Portabilité

La portabilité consiste à adapter le système d'exploitation aux capteurs, selon les éléments électroniques les constituant. Contiki est complètement écrit en langage C, ce langage de programmation est le langage le plus répandu pour la programmation des systèmes. Le portage de Contiki est effectué pour les plateformes suivantes: exp5438, z1, wismote, avr-raven, avr-rcb, avr-zigbit, iris, micaz, redbee-dev, redbee-econotag, mb851, mbxxx, sky, jcreate, sentilla-usb, msb430, esb, avr-atmega128rfa, cc2530dk, sensinode ainsi que sur apple2enh, atari, c128, c64[25].

Interopérabilité

L'interopérabilité d'un capteur est le fait de pouvoir communiquer avec des capteurs gérés par un système d'exploitation différent. Adams Dunkels de l'équipe scientifique suédoise présente dès 2003 uIP et lwIP permettant d'implémenter le protocole IP sur les systèmes limités en ressources tel que les capteurs. Jusque là, les capteurs utilisaient des protocoles de communication propriétaires ou alors des adaptations d'IP permettant le fonctionnement des applications mais sans offrir toutes les fonctionnalités du protocole IP. Dès la présentation de Contiki en 2004, uIP et lwIP étaient disponibles. De ce fait, les applications exécutées sur Contiki pouvaient dialoguer vers n'importe quel matériel supportant le protocol IP. L'arrivée d'IPv6 et uIPv6 sur Contiki apporte une nouvelle interopérabilité vers les matériels supportant ce protocole. Le support de 6LoWPAN permet à Contiki de communiquer avec les matériels via un réseau sans-fil suivant la norme 802.15.4. Contiki est réputé pour être un système d'exploitation robuste et mature, fournissant IPv4 et IPv6 pour les réseaux de capteurs sans fil. Selon une étude publiée en 2011, comprenant des tests d'interopérabilité entre des capteurs sous Contiki et d'autres sous TinyOS, l'interopérabilité est bien au rendez-vous, mais des efforts sont à faire pour mesurer et améliorer les performances des couches réseaux[25].

1.1.4. Les avantages et inconvénients du système Contiki.

L'architecture hybride du noyau Contiki autorise deux modes de fonctionnement soit multitâche, soit basé sur les événements. A ce titre, elle permet à ce système d'offrir plusieurs solutions pour répondre au, plus près, aux contraintes de l'application supportée, un mode pouvant être plus performant que l'autre. Cela constitue le principal avantage de ce système d'exploitation[11].

Cependant, le noyau Contiki reste, nativement, un système d'exploitation basé sur les événements. Pour obtenir le mode multitâche, une bibliothèque doit être installée. Les fonctions associées à cette bibliothèque n'accèdent pas directement à l'ensemble des ressources du capteur sans fil. Elles doivent, dans certains cas, faire appel à la partie du noyau dédié à la gestion des événements. Cette structure à deux niveaux a pour conséquence une dégradation des performances du système quand le mode multitâche est activé[11].

1.2. Un simulateur réseau pour Contiki : Cooja

COOJA [28] est l'acronyme de Contiki OS Java Simulator.

Pour développer les programmes au sein de Contiki, le système met à disposition un simulateur réseau appelé Cooja. Le logiciel permet d'émuler des nœuds et de charger un programme compilé. Ceci est particulièrement utile pour tester les programmes avant de les mettre dans la mémoire flash des nœuds réels, puisque le logiciel simule les conditions d'exécution et de mémoire de la plateforme TI MSP430. Les données collectées provenant du sink via sa sortie standard peuvent être enregistrées dans des fichiers ou lus par des logiciels qui peuvent par la suite traiter et présenter les données à l'utilisateur. On peut par exemple citer le logiciel collect-view intégré dans Contiki qui permet de visualiser les valeurs des capteurs et des données de supervision du réseau. En revanche, Cooja a un intérêt limité si l'on veut tester des algorithmes de géolocalisation basés sur le RSSI. En effet, le logiciel utilise un modèle linéaire pour simuler ces valeurs en fonction de la distance, alors qu'en réalité, le modèle est logarithmique. De plus, l'environnement simulé ne reflète pas la réalité en raison de la non prise en compte des perturbations engendrées par les murs, sols...

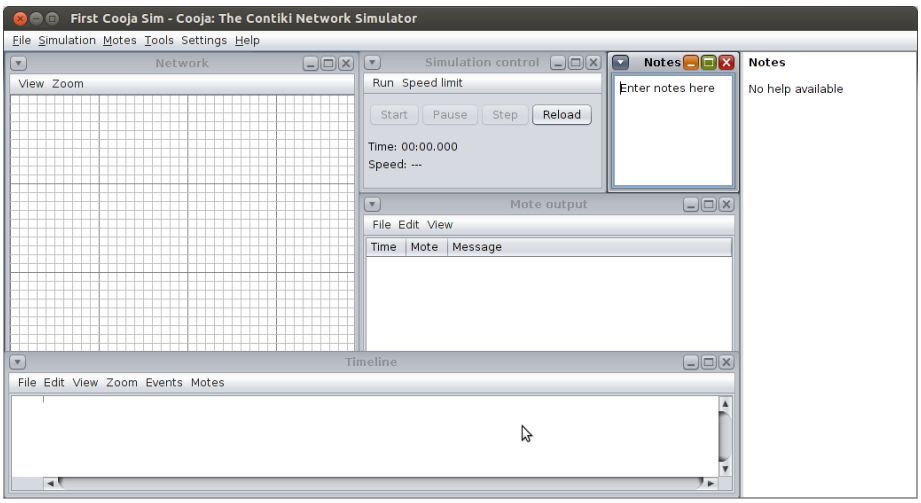


Figure III.4 : Interface de Cooja

- **Fenêtre de simulation**

Dans une simulation nous avons plusieurs fenêtres :

- La fenêtre Timeline : en bas de l'écran, nous affiche tous les événements de communication dans la simulation dans le temps, très pratique pour comprendre ce qui se passe dans le réseau.
- La fenêtre Network : en haut à gauche de l'écran, nous montre tous les nœuds dans le réseau simulé.
- La fenêtre Mote Output, sur le côté droit de l'écran, nous montre toutes les impressions port série de tous les nœuds.
- La fenêtre Notes en haut à droite est l'endroit où nous pouvons mettre des notes pour notre simulation.
- La fenêtre Simulation control : est où nous pouvons lancer, mettre en pause et charger de notre simulation.

1.3. Configuration Matérielle.

La simulation a été réalisée sur un ordinateur TOSHIBA dont la configuration est la suivante :

Processeur	Intel Core i5-3230M CPU 2.60GHz, 2.60GHz
Mémoire	4 Go DDA3
Disque dur	500 Go
Carte graphique	Intel HD Graphics 4000 NVIDIA GeForce 710M

Tableau III-1 Configuration Matérielle

2. Présentation des différents scénarios.

Notre travail consiste à étudier le temps de délivrance de l'information ainsi que la consommation d'énergie dans les réseaux de capteurs sans fil et cela en comparant les réseaux utilisant l'agrégation de Bout en Bout (End-to-End) et ceux utilisant l'agrégation en Saut par Saut (Hop-by-Hop).

Nous avons sécurisé l'agrégation des données en utilisant un algorithme de chiffrement pour chaque approche.

Durant notre phase de simulation, nous avons réalisé plusieurs topologie de réseaux de capteurs de taille variante de 5 à 50 nœuds qui sont distribués aléatoirement dans une zone de taille 350x350. La station de base est placée au coin de la zone de déploiement et elle n'est en contact direct qu'avec le Cluster Head.

Nous avons utilisé les mêmes paramètres de simulations pour les deux approches pour en déduire les résultats dans le même environnement.

En ce qui concerne la consommation d'énergie, le Cluster Head est le nœud étudié dans les différents scénarios de simulation puisqu'il s'occupe de l'agrégation des données provenant des différents nœuds collecteur vers la station de base.

Paramètre	Valeur
Taille de la zone de simulation	350m X 350m
Rayon d'émission	160m
Nombre de nœuds	5, 10, 20, 30, 40, 50
Durée de la simulation	3 minutes (180 secondes)
Déploiement des Nœuds	Aléatoire
Densité des Nœuds	Aléatoire
Nombre de clusters	1
Type de Nœuds	1 Station de Base 1 Cluster Head Plusieurs nœuds collecteurs
Approche Etudié	1 Algorithme qui se base sur le principe de End to End 1 Algorithme qui se base sur le principe de Hop by Hop
Durée d'attente avant l'envoie	10seconde

Tableau III-2: Paramètres de la simulation.

3. Architecture du réseau.

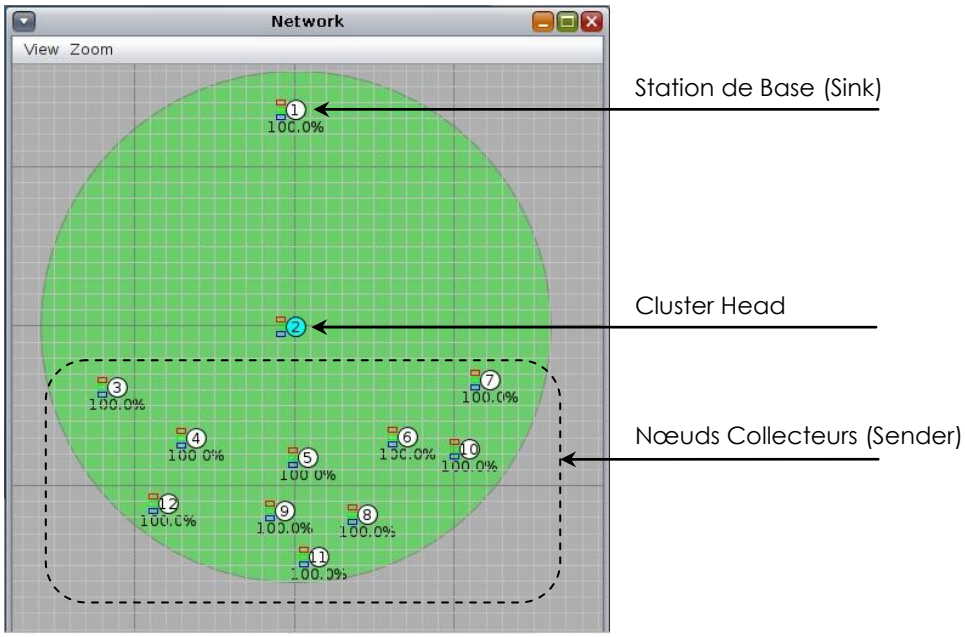


Figure III.5: Architecture Réseaux (Exemple 10 Noeuds)

3.1 Approche End-to-End.

Dans cette approche nous avons utilisé trois types de capteur : un capteur ordinaire collecteur d'information (Sender), un Cluster Head agrégateur et une Station de Base (Sink).

3.1.1. Nœud collecteur (Sender).

Il s'occupe de la détection de la température dans son environnement, chiffre cette donnée en utilisant deux clés de chiffrement (k2, M) puis l'envoi au Cluster Head. La clé k2 est obtenue en utilisant l'ID du nœud collecteur selon le code qui suit :

```
id_loc=rimeaddr_node_addr.u8[0];
k=id_loc*800+10000;
k2=0;
for (i=1;i<=k;i++){k2=k2+i;}
```

La clé M est partagée entre la station de base et le nœud collecteur.

Le chiffrement est effectué par le code qui suit :

```
to_send_msg->Temp_snd=(tmp_loc+k2)%M;
```

avec tmp_loc la température locale captée.

3.1.2. Cluster Head.

Il s'occupe de recevoir les données transmises par les nœuds Sender sans les déchiffrer, calcule leur somme chiffre le résultat en utilisant deux Clés partagées avec la station de base (Khd , Ms) comme le montre le code qui suit :

```
tmp_loc=(tmp_rc+k_hd)%MS;
```

avec tmp_loc la température à envoyer et tmp_rc la température reçue (somme)

3.1.3 Station de base (Sink).

La Station de Base s'occupe du déchiffrement des données pour en déduire la température moyenne dans le réseau.

Premièrement, elle déchiffre la somme reçue (Temp_CS) depuis le Cluster Head en utilisant les deux clés partagées (Khd, MS) comme indiqué dans l'instruction du code qui suit :

```
tmp_cs=((recv_msg->Temp_CS)+((k_hd/MS)*MS)-(k_hd));
```

Ensuite, elle déchiffre la somme tmp_cs pour en déduire la température moyenne dans le réseau en utilisant les deux clés partagées (k2, M) entre elles et chaque nœud Sender comme le montre le code :

```
tmp= tmp_cs+(y*M)-(kt)/ndt
```

avec ndt nombre de nœuds ayant transmis leurs températures détectées

3.2. Approche Hop-by-Hop.

Dans cette approche nous avons également utilisé les mêmes trois types de capteurs : Sender, Cluster Head et Sink.

3.2.1. Nœud Sender.

Le Sender chiffre la température captée en utilisant toujours deux clés de chiffrement (k2, M) mais qui sont partagées avec le Cluster Head et le Sink puis l'envoi à ce dernier.

La clé k2 est obtenue en utilisant l'ID du nœud collecteur selon le code qui suit :

```
Temp_snd=(tmp_loc+k2)%M;
```

Avec tmp_loc la température locale captée.

3.2.2. Cluster Head.

Après avoir reçu une donnée transmise par un nœud Sender, le cluster Head déchiffre cette donnée et calcule la somme des données reçues déchiffrées (valeur réelles captées) comme montré dans le code qui suit :

```
tmp_rc_sm=tmp_rc_sm+((recv_msg->Temp_rec)+(y*M)-(k2));
```

Avec Temp_rec la température chiffrée reçu d'un nœud Sender.

Après, le Cluster Head chiffre la moyenne des données reçues par son cluster en utilisant deux clés partagées avec la station de base (Khd , Ms) et l'envoie au Sink comme le montre le code qui suit :

```
tmp_rc=tmp_rc_sm/nbr;
tmp_cs=(tmp_rc+k_hd)%MS;
to_send_msg->Temp_snd=tmp_cs;
```

Avec nbr le nombre des nœuds dans le cluster ayant envoyé des données au cluster Head.

3.2.3. Station de base (Sink).

Dans cette approche, la station de base reçoit la température moyenne chiffrée envoyée par le cluster Head, elle déchiffre la donnée reçue (Temp_CS) en utilisant les deux clés partagées (Khd , MS) comme montré dans le code qui suit :

```
tmp=((recv_msg->Temp_CS)+(y*MS)-(k_hd));
```

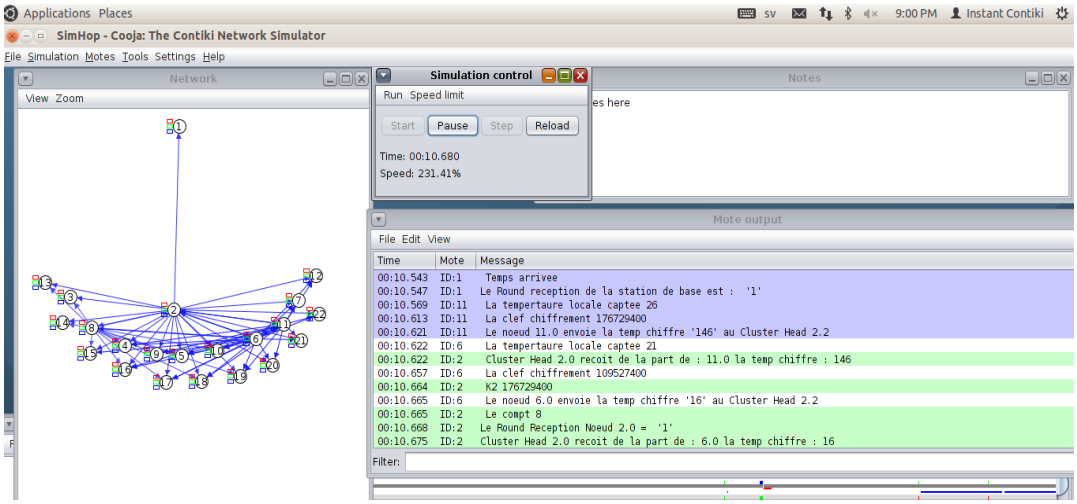


Figure III.6: Exemple d'une partie de la simulation avec 20 Nœuds.

4. Métrique de simulation

Notre étude est basée sur deux métriques très importantes, voire même critiques dans les réseaux de capteurs sans fils qui sont : **le temps de délivrance** de l'information et **la consommation d'énergie**. Le temps de délivrance est critique dans une zone ou domaine surveillé par un WSN, un simple retard peut causer d'importants dégâts. La consommation d'énergie est un facteur vital dans un WSN, une fois l'énergie épuisée le réseau ne sera plus fonctionnel.

4.1 Délai de délivrance.

Dans notre étude, le délai de délivrance d'information est la différence entre le temps d'affichage de température moyenne dans le réseau au Sink et celui de l'envoie de la valeur de premier capteur Sender dans le cluster.

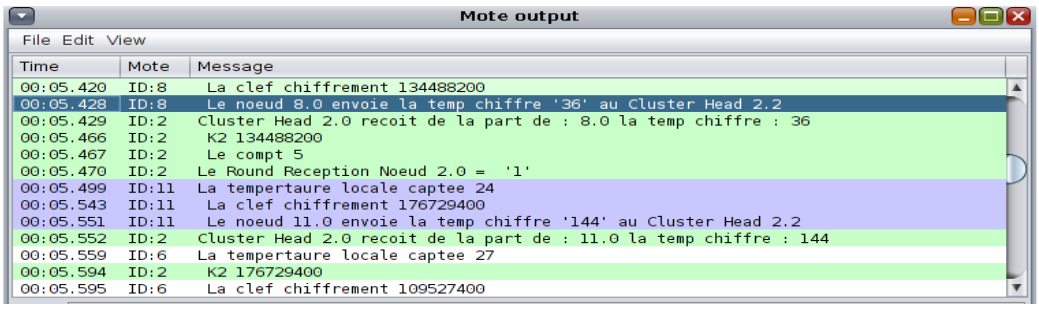


Figure III.7: Le Temps de l'envoie de la température par le neous Sender

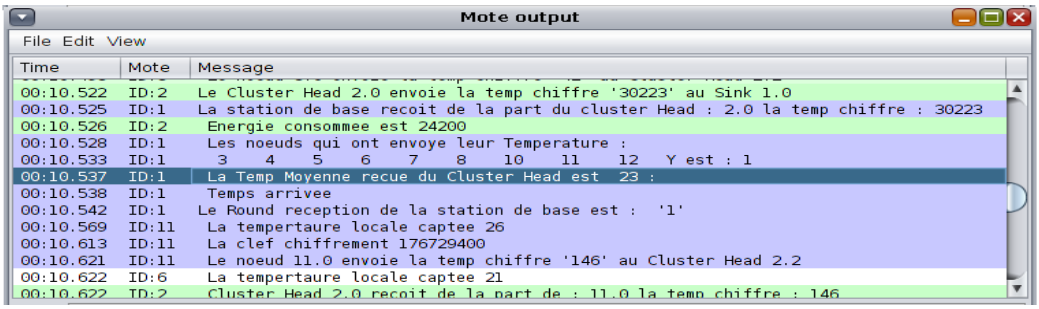


Figure III.8: Le Temps de la réception de la température par la Station de Base

Les tableaux ci dessous montrent les résultats obtenus dans les deux approches.

Nombre de Nœuds	Temps de Délivrance (ms)	
	End to End	Hop by Hop
5	5254	4933
10	5590	5109
20	6498	5114
30	7266	5203
40	7369	5262
50	7396	5199

Tableau III-3: Le Temps de Délivrance.

Le graphe suivant illustre la différence des résultats entres les deux approches.

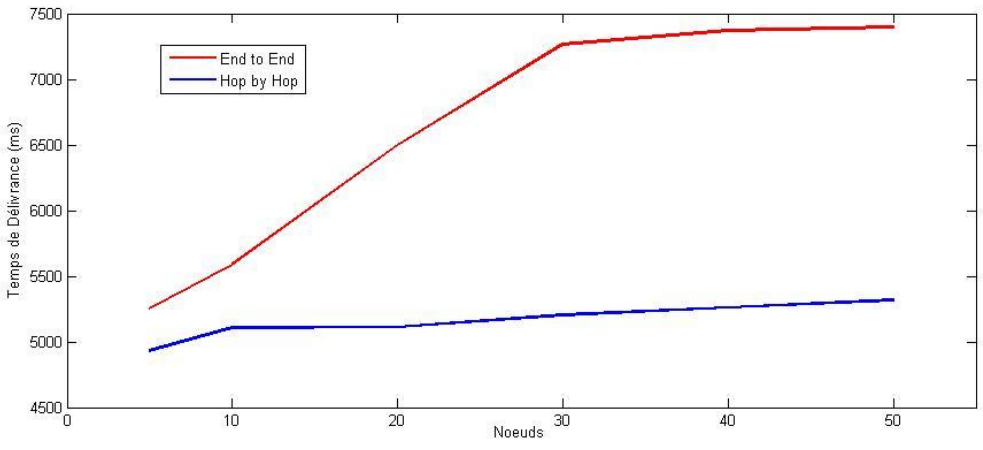


Figure III.9: Temps de Délivrance

4.2 Consommation d'énergie

Le Cluster Head représente l'axe de notre étude de consommation d'énergie dans le WSN puisqu'il représente une passerelle entre le Sink et le Sender et effectue des opérations plus importantes qu'un nœud ordinaire, nous avons utilisé la fonction **energest** dans "sys/energest.h" pour afficher la quantité d'énergie consommée par le Cluster Head.

```
#include "sys/energest.h"
.
.
 niv_energ = energest_type_time(ENERGEST_TYPE_CPU);
```

Les tableaux suivants résument tous les résultats obtenus durant les simulations le temps est en millisecondes et l'énergie consommée est en milli-joule.

5 Nœuds		10 Nœuds		20 Nœuds		30 Nœuds		40 Nœuds		50 Nœuds	
Temps	Energie	Temps	Energie	Temps	Energie	Temps	Energie	Temps	Energie	Temps	Energie
10527	12270	10527	13007	10527	15976	10527	19256	10531	22487	10531	25153
20543	24735	20543	27471	20543	33748	20543	41447	20543	48791	20543	56001
30558	37205	30558	41981	30558	52179	30558	64598	30558	75714	30558	86262
40574	49673	40574	56484	40574	70601	40574	87488	40575	102345	40575	116943
50590	62140	50590	70974	50590	89012	50590	110316	50590	128618	50590	147907
60605	74609	60605	85480	60606	107427	60606	133699	60606	154938	60606	178232
70621	87040	70621	99949	70621	125791	70621	156534	70621	180680	70621	208934
80637	99506	80637	114442	80637	144171	80637	179094	80637	206468	80637	239358
90653	111978	90652	128962	90653	160729	90653	199845	90652	230447	90653	267980
100668	124457	100668	143474	100668	177340	100668	220355	100668	254182	100668	295769
110684	136935	110684	157985	110684	195772	110684	243224	110684	280282	110684	326506
120699	149420	120699	172508	120699	214220	120699	266142	120700	306433	120700	357292
130715	161898	130715	186985	130715	232620	130715	289021	130715	332542	130715	388038
140731	174376	140731	200929	140731	250487	140731	311038	140731	357791	140731	417920
150746	186866	150746	214589	150746	268059	150746	333075	150746	383059	150746	447829
160762	199343	160762	228247	160762	285635	160762	355134	160762	407711	160762	477759
170778	211823	170778	241908	170778	303137	170778	376436	170778	432872	170778	506924
180793	224312	180793	255570	180793	320711	180793	398479	180793	458143	180793	536836

Tableau III-4 : Les résultats de la consommation d'énergie pour le End to End

5 Nœuds		10 Nœuds		20 Nœuds		30 Nœuds		40 Nœuds		50 Nœuds	
Temps	Energie	Temps	Energie	Temps	Energie	Temps	Energie	Temps	Energie	Temps	Energie
10526	17519	10526	24200	10526	44798	10526	33619	10590	39707	10600	45201
20533	35244	20533	50300	20533	91820	20533	80107	20604	90852	20613	123045
30541	52976	30541	76456	30542	138919	30542	131645	30612	142623	30637	187630
40549	70702	40549	102601	40549	186009	40549	180702	40620	197249	40665	227895
50557	88431	50557	128734	50557	233099	50557	231564	50627	247416	50709	265240
60565	106159	60565	154887	60565	280207	60565	281392	60635	313034	60565	308599
70573	123899	70573	181049	70573	327288	70573	335597	70643	372526	70573	367983
80580	141636	80580	207137	80580	368546	80580	386606	80651	428947	80580	432120
90588	159381	90588	233303	90588	404677	90588	446386	90658	482653	90588	498741
100596	177118	100596	259448	100596	451713	100596	510154	100667	542257	100596	568896
110604	194858	110604	285602	110604	498822	110604	566707	110674	592885	110604	631847

120612	212603	120612	311761	120612	545937	120612	624614	120682	653993	120612	701932
130620	230344	130620	337891	130620	593015	130620	688260	130690	720539	130620	772113
140627	248081	140627	364051	140627	640133	140627	752077	140698	787564	140627	842312
150635	265830	150635	390221	150635	687272	150635	815931	150705	854617	150635	912545
160643	283568	160643	416390	160643	734419	160643	879764	160713	921660	160643	982768
170651	301307	170651	442558	170651	781506	170651	943478	170721	988576	170651	1052865
180658	319056	180658	468738	180658	828617	180659	1004751	180729	1053058	180659	1120531

Tableau III-5: Les résultats de la consommation d'énergie pour le Hop-by-Hop

Les graphes suivants illustrent les différents résultats de consommation d'énergie dans le cluster Head dans 12 simulations, 6 pour chaque approche, variant de 5 à 50 le nombre de nœuds.

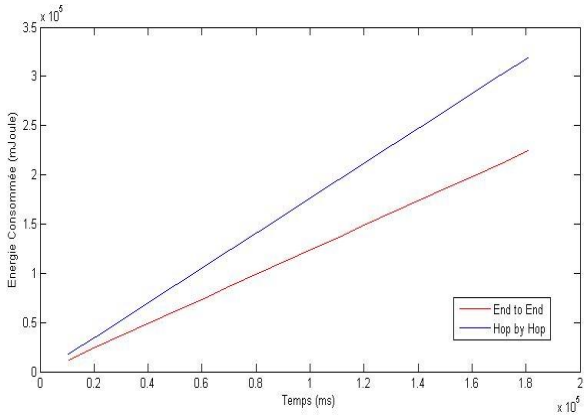


Figure III.10: Consommation d'énergie Avec 5 Nœuds.

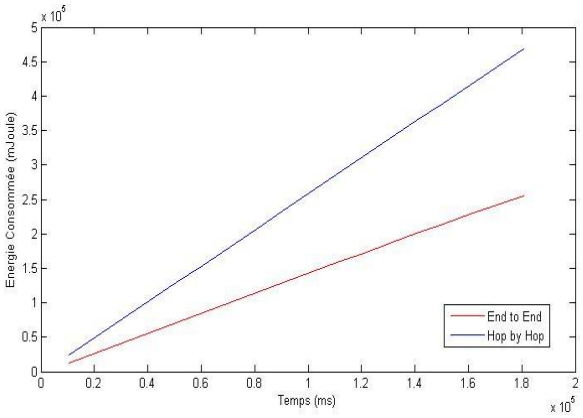


Figure III.11: Consommation dans le Temps Avec 10 Nœuds.

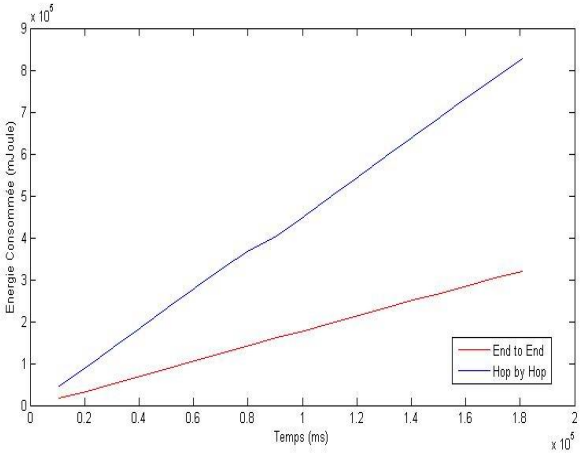


Figure III.12: Consommation dans le Temps Avec 20 Nœuds.

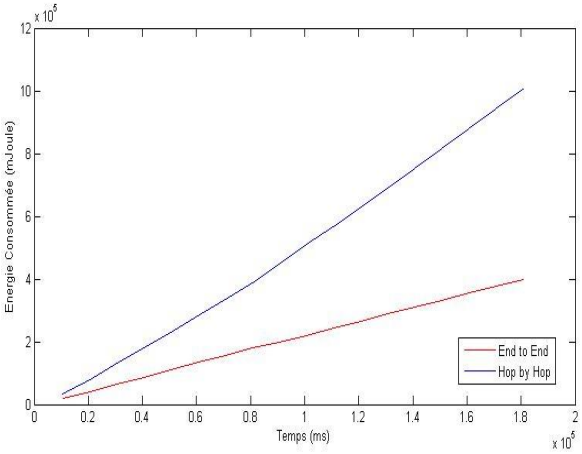


Figure III.13: Consommation dans le Temps Avec 30 Nœuds.

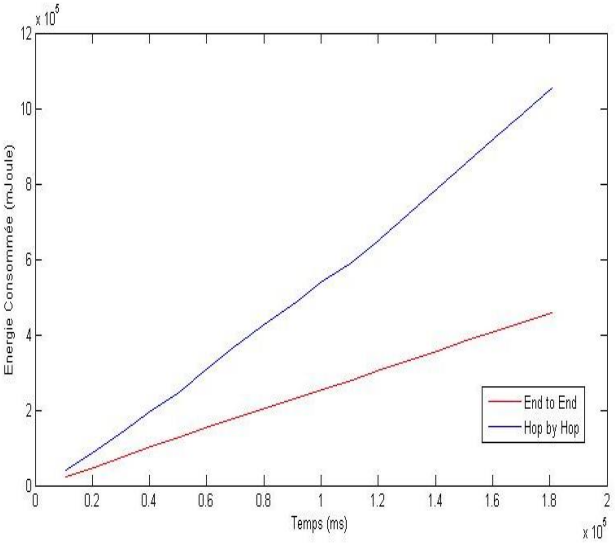


Figure III.14: Consommation dans le Temps Avec 40 Nœuds.

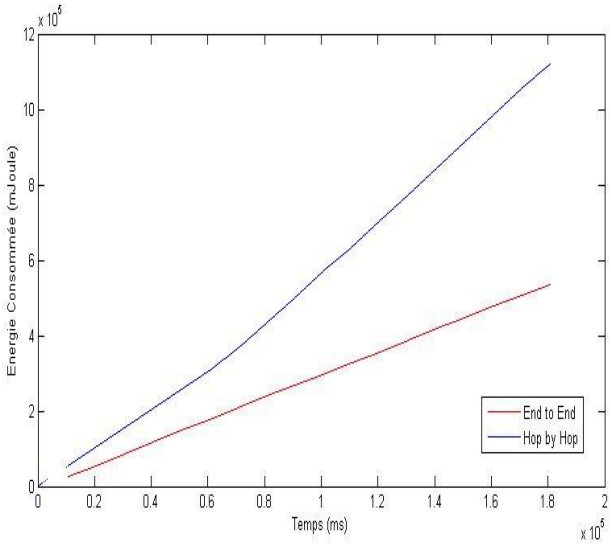


Figure III.15: Consommation dans le Temps Avec 50 Nœuds.

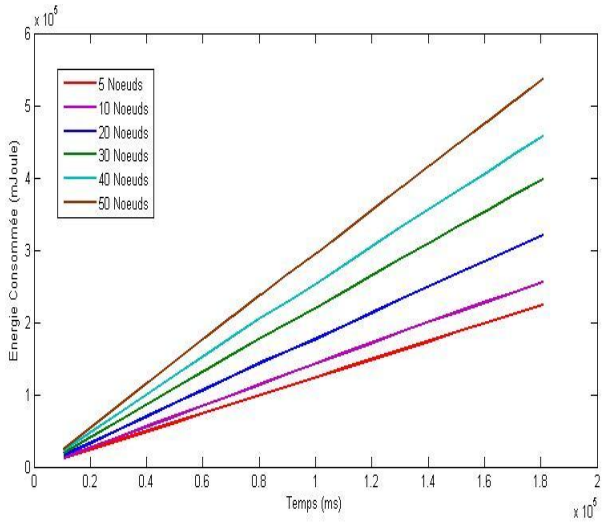


Figure III.16: Energie consommée par le Cluster Head utilisant le End to End.

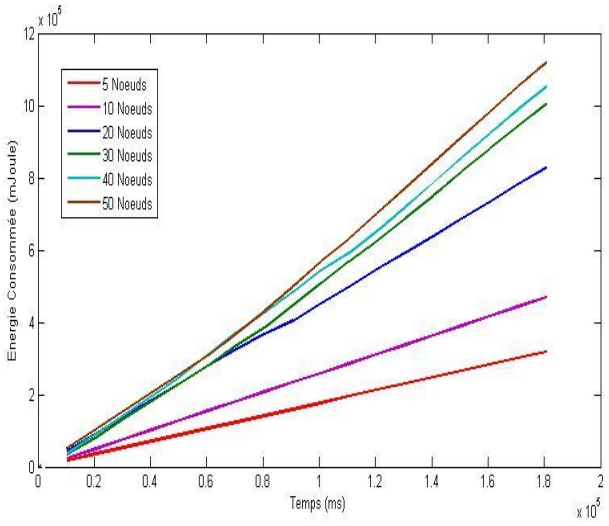


Figure III.17: Energie consommée par le Cluster Head utilisant le Hop by Hop.

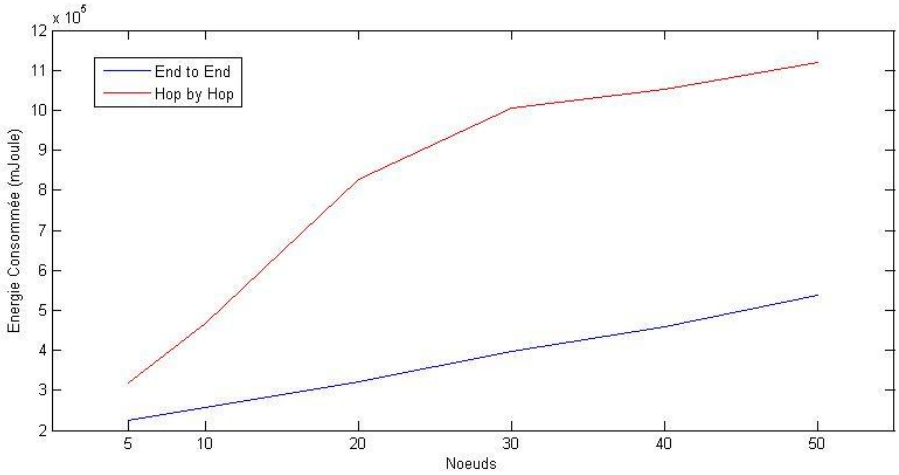


Figure III.18: Energie Consommée par le Cluster Head Après 3 Minutes.

5. Discussion des Résultats

5.1 Temps de délivrance

5.1.1. End-to-End :

Dans l'approche End-to-End, le cluster Head accumule les données chiffrées reçues par les Senders et les transmet au Sink après les avoir chiffrées à nouveau. Ce dernier s'occupe du déchiffrement du message reçu à deux niveaux pour enfin déduire la température moyenne dans le réseau. Le temps de délivrance d'information dans cette approche est plus

élevé puisque le déchiffrement au niveau de la station de base est plus complexe. Ce temps augmente en augmentant le nombre de nœuds dans le cluster. Plus il y a de nœuds, plus le traitement de déchiffrement au Sink est plus important. (Figure III.9)

5.1.2. Hop-by-Hop :

Dans l'approche Hop-by-Hop, le cluster Head déchiffre chaque donnée reçue par les Senders et calcule la moyenne des données reçue en claire, puis chiffre cette moyenne et l'envoie au Sink, Ce dernier déchiffre cette moyenne et affiche le résultat. Le temps de délivrance dans cette approche est moins important que la précédente puisque le cluster Head calcule la moyenne de son cluster et l'envoie au sink, ce qui allège l'opération de déchiffrement au niveau de ce dernier. On remarque que le temps de délivrance augmente en augmentant le nombre de nœuds dans le cluster, du fait que ce dernier a besoin de plus de temps pour déchiffrer toutes les données reçues des Senders. Mais cette augmentation demeure moins importante qu'en End-to-End. (Figure III.9)

5.2 Consommation d'énergie

En termes de consommation d'énergie, on remarque un écart important entre les deux approches. L'augmentation des nœuds dans les deux cas mène à accroître la consommation du Cluster Head, car l'ajout de nouveaux nœuds Senders mène à un trafic radio plus important. Mais cette hausse de consommation est différente entre les deux approches.

5.2.1. End-to-End :

Dans cette approche, le cluster Head envoie la somme des données reçues sans avoir besoin de les déchiffrer, il suffit de calculer la somme de ces données et les chiffrer pour les envoyer au Sink et ce dernier s'occupe des opérations de déchiffrement. Dans les 6 simulations on remarque que la consommation d'énergie avec 5 nœuds est de $2x10^5$ mJ (Figure III.10) et atteint presque $6x10^5$ mJ en augmentant le nombre de nœuds à 50 (Figure III.15). Cette hausse de consommation n'est influencée que par la multiplication des opérations de réception de données par les Senders. Le calcul de la somme à envoyer ne change pas dans les 6 simulations, donc il ne sera pas important en termes de consommation d'énergie.

5.2.2. Hop by Hop :

Dans l'approche Hop by Hop, le cluster Head est le nœud clé dans le réseau, il s'occupe de la réception et du déchiffrement de chaque donnée transmise par un Sender, il calcule la clé k2 de chaque Sender, réalise le déchiffrement des données reçues, calcule la moyenne de ces données et enfin chiffre et envoie le résultat au Sink. Dans les 6 simulations on remarque que la consommation d'énergie avec 5 nœuds est de $3x10^5$ mJ (Figure III.10) et elle ne cesse de s'accroître pour arriver à $8x10^5$ mJ avec 20 nœuds (Figure III.12) pour enfin atteindre $12x10^5$ mJ en augmentant le nombre de nœuds à 50 (Figure III.15). Dans cette approche, l'augmentation de nœuds dans le réseau n'influence pas uniquement le trafic radio mais aussi bien le traitement de déchiffrement au niveau du cluster Head, plus on a de nœuds, plus on a de calcul de clés et de traitement de déchiffrement pour chaque donnée reçue.

5.2 Critique comparative des deux approches

D'après les résultats obtenus, on peut constater que chaque approche a ses avantages et ses inconvénients. Dans l'approche End-to-End, la collecte d'information est plus sécurisée puisque le cluster Head envoie des données chiffrées au Sink sans avoir besoin de les lire ou les déchiffrer. La consommation d'énergie en End-to-End est moins importante (Figure III.16), ce qui représente un avantage pour le WSN. Mais en termes de temps de délivrance, le Sink doit faire plusieurs opérations de déchiffrement et cela va retarder l'affichage du résultat. En Hop-by-Hop, le cluster Head doit déchiffrer toute donnée reçue ce qui représente un inconvénient de sécurité. La consommation d'énergie dans cette approche est beaucoup plus importante (Figure III.17), ce qui peut influencer le bon fonctionnement du WSN en cas d'épuisement d'énergie. Mais en termes de temps de délivrance, le Sink ne fait que déchiffrer la moyenne reçue par le cluster Head et cela permettra une récupération plus rapide du résultat.

Conclusion

Dans ce chapitre nous avons présenté une étude comparative en termes de temps de délivrance et de consommation d'énergie dans un réseau de capteur sans fil selon les deux approches d'agrégation sécurisée de données : End to End et Hop by Hop.

La sécurisation des données se réalise en utilisant deux clés de chiffrement selon la fonction suivante :

Tmp_CS=(Tmp_loc+k2)%M

Avec Tmp_CS : valeur chiffrée à envoyer, Tmp_loc : Température locale captée, k2 : clé calculée a partir de l'ID de chaque nœud Sender et M une clé partagée soit entre le Sender et le Sink en End-to-End ou entre le Sender et le cluster Head en Hop-by-Hop.

Le déchiffrement de fait à l'aide de la fonction suivante :

Tmp=Tmp_rec+(y*M)-(Kt));

Avec Tmp : valeur déchiffrée, Tmp_rec : Température chiffrée reçue, y :somme des division entière de chaque clé d'un Sender par M , M une clé partagée soit entre le Sender et le Sink en End-to-End ou entre le Sender et le cluster Head en Hop-by-Hop. Kt : ensemble des clés k2 des Sender ayant envoyé leurs températures.

L'étude comparative des deux approches en utilisant les deux métriques : temps de délivrance et consommation d'énergie a montré que chacune des approches possède des avantages et des inconvénients. Dans un domaine où l'économie d'énergie est demandée, la solution en End-to-End est plus opérationnelle. En revanche, si le temps d'affichage du résultat final est critique, une solution en Hop-by-Hop sera plus appropriée.

Conclusion Générale

L'installation d'un réseau de capteurs sans fil est un projet nécessitant une étude du domaine et de la zone d'implémentation et une connaissance précise des besoins nécessaires et des résultats attendus. Le bon déploiement des nœuds est très important pour le bon fonctionnement du RCSF, donc le choix d'une topologie qui respecte les conditions géographiques et techniques est critique.

En plus de la bonne conception du réseau et du déploiement précis des capteurs, l'agrégation de données représente une technique très utile pour économiser les ressources énergétiques et faciliter la collecte d'informations. La sécurité de l'agrégation de données est nécessaire dans un RCSF, des données en claire peuvent facilement être interceptées ou même falsifiées.

Durant notre modeste travail, nous avons évalué les deux approches de sécurisation de l'agrégation des données : de Bout-en-Bout (End-to-End) et de Saut-par-Saut (Hop-by-Hop). Nous avons évalué la performance du RCSF en termes de deux métriques : temps de délivrance de l'information et consommation d'énergie.

Nous avons constaté que chaque approche possède des avantages et des inconvénients. En End-to-End, la consommation d'énergie est moins importante, ce qui représente un grand avantage pour les RCSF déployés dans des endroits avec des conditions géographiques difficiles ou des zones inaccessibles (par exemple surveillance d'un phénomène naturel) mais en revanche le temps de délivrance d'informations est plus lent.

En Hop-by-Hop, la consommation d'énergie est beaucoup plus importante, ce qui influence directement les ressources énergétiques du capteur, donc cette solution est plus adaptable aux environnements où on peut intervenir pour remplacer les batteries des capteurs (Domaine agricole par exemple), en revanche le temps de délivrance d'information est plus rapide.

Du coté sécurité, l'approche End-to-End est plus fiable en termes d'agrégation de données, l'agrégateur n'a aucun accès aux données chiffrées, il ne fait qu'acheminer les données reçues à la station de base. L'agrégateur dans l'approche Hop-by-Hop déchiffre chaque donnée reçue d'un capteur, l'agrégation est effectuée sur des données en claire. Cette

solution est peu fiable si l'environnement de déploiement n'est pas sûr, il suffit qu'un agrégateur soit compromis pour que toutes les données d'une zone soient fausses.

Donc, comme perspective, l'étude de sécurité des nœuds dans les deux approches sera intéressante, en supposant qu'un ou plusieurs capteurs seront compromis ou même en supposant que l'agrégateur lui même est compromis et prévoir des solutions pour le RCSF contre les attaques de sécurité. Aussi, la combinaison entre les deux approches pour en profiter des avantages de chacune pourra beaucoup améliorer la performance du RCSF

Bibliographie

[1] S. Athmani - Protocole de sécurité Pour les Réseaux de capteurs Sans Fil– Thèse de Magistère en Informatique, Département d'Informatique, Université Hadj Lakhder de Batna, Juin 2010.

[2] N. Norbet. – Du signal à l'information : le capteur intelligent Exemples industriels et en médecine. Grenoble: TIMC-IMAG, 2002.

[3] S. Hadjadj – Système de supervision dans les réseaux de capteurs sans fil – Mémoire de Master, Département de Math et Informatique, Université Tahri Mohamed de Bechar, 2014-2015.

[4] N. Labraoui, – La sécurité dans les réseaux sans fil ad hoc–, Thèse de Doctorat, Département d'Informatique, Université de Tlemcen, 2012.

[5] A. Bendjeddou, – Prolongation de la Durée de Vie des Batteries dans les Réseaux de Capteurs Sans Fil (RCSF) – Thèse de Doctorat 3ème Cycle, Département d'Informatique, Université Badji Moukhtar de Annaba– 2014-2015.

[6] CT.Kone – Conception de l'architecture d'un réseau de capteurs sans fil de grande dimension – Thèse de Doctorat 3ème Cycle, Département d'Automatique et Production Automatisée, Université Hanry Poincaré de Nancy I– 18 octobre 2011.

[7] B.A. Bensaber, – Introduction à la sécurité des réseaux de capteurs sans fil– Département de Mathématique Informatique, Université de Québec à Trois Rivières, Juin 2013.

[8] T. Watteyne: –Energy-Efficiency Self-Organisation for Wireless Sensor Network–, Doctorate Thesis, Institut National des Sciences Appliquees de Lyon, 2008.

[9] L. Eschenauer, V. D. Gligor: –A Key Management Scheme for Distributed Sensor Networks–, ACM CCS, 2000.

[10] M. Abdallah, – Réseaux de capteurs : localisation, couverture et fusion de données–, Thèse de Doctorat, Université de Franche-Comté, Novembre 2008.

[11] G.D Sousa, « Etude en vue de la réalisation de logiciels bas niveau dédiés aux réseaux de capteurs sans fil : microsystème de fichiers», Thèse de Doctorat, Département d'Informatique, Université Blaise Pascal-Clermont II, Octobre 2008.

[12] M.J. Handy, M. Haase, D. Timmermann: – Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection–, Fourth IEEE Conference on Mobile and Wireless Communications Networks, Stockholm, erschienen in Proceedings, 2002.

[13] Y. Challal, –Réseaux de Capteurs Sans Fils–, support de cours -SIT60, Vol. 103, Novembre 2008.

[14] S.Ed. Benbrahim – Défense Contre L'attaque D'analyse De Trafic Dans Les Réseaux De Capteurs Sans Fil (WSN) –Thèse de maitrise en Science Appliquées – Département De Génie Informatique Et Génie Logiciel Ecole Polytechnique De Montréal

[15] D. Martins – Sécurité dans les réseaux de capteurs sans fil Stéganographie et réseaux de confiance – Thèse Doctorat, L'U.F.R. Des Sciences Et Techniques de l'université de Franche-Comté – 2010.

[16] S.A.H. Sedjelmaci– Mise En Œuvre de Mécanismes de Sécurité Bases sur les Ids pour Les Réseaux de Capteurs Sans Fil –, Thèse de Doctorat, Faculté de Technologie, Université de Tlemcen, 2012.

[17] Y. Challal, – Sécurité de l'Internet des Objets : vers une approche cognitive et systémique– Rapport Au vu d'obtenir le diplôme d'Habilitation à Diriger des Recherches, Université de Technologie de Compiègne, 2012.

[18] K. Wua, D. Dreefa, B. Sunb, Y. Xiao, – SDAP Secure data aggregation without persistent cryptographic operations– in :wireless sensor networks, Ad Hoc Networks– Volume 5– January 2007.

[19] L. Hu, D. Evans, – SAWN : Secure aggregation for wireless networks–, in: Workshop on Security and Assurance in Ad Hoc Networks, January 2003.

[20] A. Mahimkar, T. Rappaport, – SecureDAV: A secure data aggregation and verification protocol for sensor networks–, in: Proceedings of the IEEE Global Telecommunications Conference, Volume 4 – December 2004.

[21] H. Alzaid, E. Foo, J. Nieto, – RSDA: Reputation-based secure data aggregation in wireless sensor networks–, in: International Conference on Parallel and Distributed Computing, Applications and Technologies, December 2008.

[22] C. Castelluccia, E. Mykletun, G. Tsudik, – CMT : Efficient aggregation of encrypted data in wireless sensor networks–, in: Mobile and Ubiquitous Systems: Networking and Services, July 2005.

[23] R. Bista, K.J. Jo, J.W. Chang, – ASAP : A new approach to secure aggregation of private data in wireless sensor networks–, IEEE International Conference on Dependable, Autonomic and Secure Computing–, December 2009.

[24] A. Dunkels, B. Grönwall, T. Voigt « Contiki – a Lightweight and Flexible Operating System for Tiny Networked Sensors » 1st IEEE Workshop on Embedded Networked Sensors (IEEE EmNetS-I), Tampa, Florida, USA, November 2004.

[25] A. Badaoui, « Acquisition de données à distance dans les réseaux de capteurs sans fil », Mémoire de master, Département d'Informatique, Université de Tlemcen, juin 2013.

[26] A. Dunkels, O. Schmidt, T. Voigt, M. Ali « *Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems* » 4th International Conference on Embedded Networked Sensor Systems (SenSys'06), Boulder, Colorado, USA, November 2006.

[27] R. Lajara, J. Peligri-Sebastia, J. P. Solano, « Power Consumption Analysis of Operating Systems for Wireless Sensor Networks », SENSORS, 2010, p. 5809-5826.

[28] www.contiki-os.org.

Glossaire

A :

• ***Acknowledgement spoofing.*** Dans cette attaque, l'intrus tente de convaincre l'expéditeur que le lien faible est fort ou qu'un nœud mort est vivant. Par conséquent, tous les paquets qui passent par ce lien ou ce nœud seront perdus.

B :

Black holes. Dans cette attaque, l'intrus prétend être dans le plus court chemin vers la station de base ou le *cluster-head* en générant une puissance élevée de transmission. Le RSCF est vulnérable à ce genre d'attaque en raison de leur paradigme de communication, où tous les nœuds acheminent les données vers un nœud centralisé. Par conséquent, tous les paquets reçus par ce nœud malveillant seront supprimés.

C :

CAN : Convertisseur Analogique-Numérique.

L :

LEACH : *Low-Energy Adaptive Clustering Hierarchy*.

M :

MAC : Une adresse MAC (Media Access Control), parfois nommée adresse physique, est un identifiant physique stocké dans une carte réseau ou une interface réseau similaire.

MD2 : Message Digest 2.

MD5 : Message Digest 5.

S :

• ***Selective forwarding.*** Dans cette attaque, l'attaquant empêche la transmission de certains paquets. Ces derniers seront par la suite supprimés par ce nœud malveillant.

• ***Spoofed, Altered, et Replayed packets.*** L'attaquant surveille les transmissions, intercepte les paquets, puis modifie les informations de routage et les réutilise pour générer des faux messages d'erreur.

• ***Sinkhole et Hello flood.*** La caractéristique commune entre les deux attaques, est que le noeud malveillant va convaincre ses voisins que c'est le nœud le plus proche de la station de base ou du *cluster-head* en utilisant une puissance de transmission élevée. Par conséquent tous les paquets reçus seront modifiés et transmis à la station de base ou à l'utilisateur.

SAWN : Secure Aggregation for Wireless Networks.

SHA-1 : Secure Hash Algorithm.

W :

• ***Wormholes.*** Connues aussi sur le nom de *tunneling.* Dans cette attaque, un adversaire peut recevoir des messages et les rejouer dans différentes parties à l'aide d'un tunnel entre deux noeuds malicieux.

WSN : Wireless Sonsor Network

Annexe

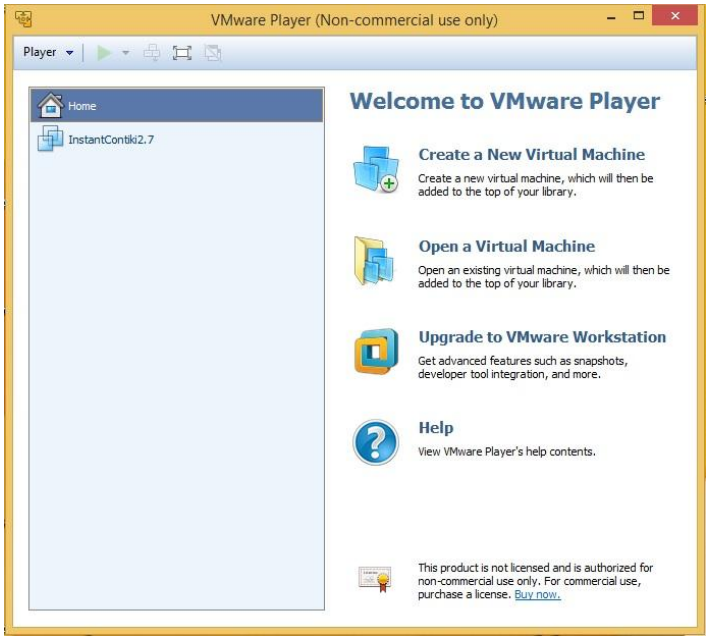
Annexe 1 : Installation de Contiki.

Contiki est un système d'exploitation pour les réseaux de capteurs sans fil. Cooja est un émulateur réseau basé sur Contiki qui permet d'exécuter des programmes sur Contiki sans avoir besoin du matériel. Pour éviter d'installer tout l'environnement de développement, nous allons utiliser une machine virtuelle (VM) nommée «Instant Contiki».

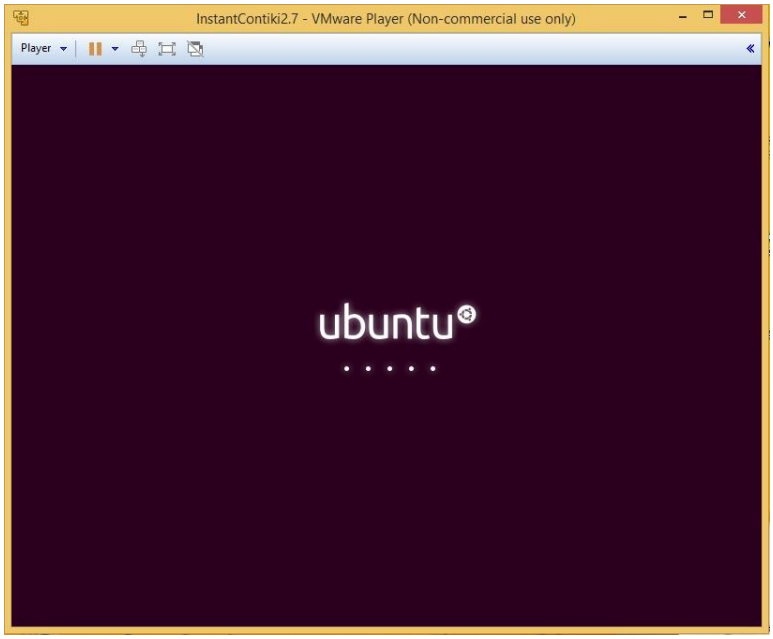
Etapes à suivre pour mettre en place la VM :

1- Télécharger la machine virtuelle nommée : « Instant Contiki 2.7 ». Le site : http://sourceforge.net/projects/contiki/files/Instant%20Contiki/ ou http://www.contiki-os.org/ Il s'agit d'un fichier de grande taille, un peu plus de 1 Gigaoctet. Une fois télécharger, décompressez le fichier et placez le répertoire décompressé sur le bureau

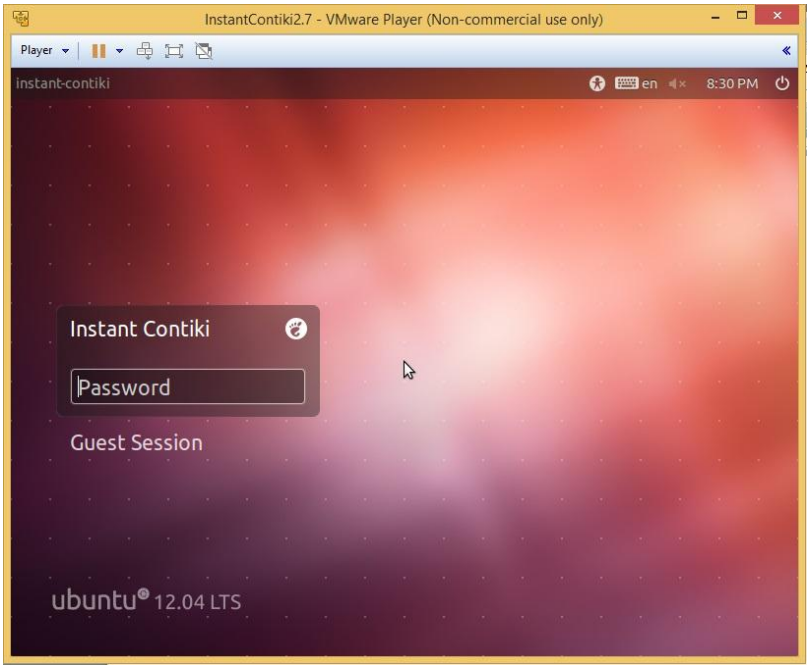
2- Pour faire tourner cette machine virtuelle, il faut télécharger un lecteur des machines virtuelles comme VirtualBox ou VMPlayer. Si vous utilisez VirtualBox, il faut vérifier que l'option « PAE/NX » est activée (Settings → System → Processor)



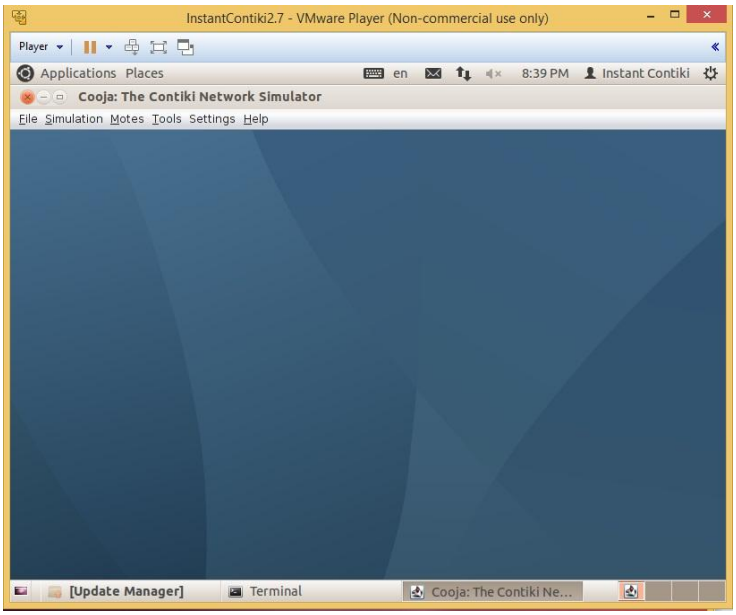
3- Une fois que vous avez installé le lecteur des machines virtuelles avec la machine Instant Contiki, vous allez avoir besoin du login et mot de passe.



Dans ce cas vous utilisez le mot de passe : user.



4- Lancer l'émulateur Cooja : Pour lancer l'émulateur "Cooja", il suffit d'aller cliquer sur l'icône Cooja qui se situe sur le bureau, il va prendre un peu de temps pour se compilé puis afficher une fenêtre bleue :



Dés que vous auriez cette fenêtre vous pouvez commencer à simuler vos topologies.

Résumé

Le domaine d'application des réseaux de capteurs ne cesse d'accroître avec le besoin d'un mécanisme de sécurité efficace. Le fait que les RCSF traitent des données très souvent sensibles, opérant dans des environnements hostiles et inattendus, la notion de sécurité est considérée comme indispensable. Cependant, à cause de la limitation des ressources et la faible capacité de calcul d'un nœud capteur, ainsi que les ressources limitée en énergie et la consommation énergétique élevée en communication, le choix d'un modèle de déploiement garantissant une sécurité pose de vrais défis.

L'agrégation des données est une technique très utilisée car elle permet de réduire le nombre de message transmis dans le réseau et par conséquent réduire la consommation d'énergie, ainsi améliorer la durée de vie du réseau.

Dans cette thèse, nous nous sommes intéressés aux problèmes de sécurité liés à l'agrégation des données dans les réseaux de capteurs sans fil. Nous avons étudiés deux approches Bout-en-Bout et Saut-par-Saut et déduire les avantages et les inconvénients de chaque approches.
Mots-clés : RCSF, capteur, agrégation de donnée, sécurité, énergie, Bout-en-Bout, Saut-par-Saut Contiki.

Abstract

The field of application of sensor networks continues to increase with the need for effective security mechanism. The fact that these networks deal with the often sensitive data, operating in hostile and unexpected environments, the concept of security is considered essential. However, because of limited resources and low computing capacity of a sensor node and the limited energy resources and the high energy consumption in communication, choosing a deployment model ensuring security is real challenges.

Aggregation of Data is a widely used technique because it reduces the number of messages transmitted in the network and thereby reduce energy consumption and improve the life of the network.

In this thesis, we focused on safety issues related to data aggregation in wireless sensor networks. We studied two approaches : End-to-End and Hop-by-Hop and deduct the advantages and disadvantages of each approach.
Keywords: WSN, sensor, aggregation of data, security, energy, End-to-End, Hop-by-Hop, Contiki.

ملخص

إن مجال تطبيق شبكات الاستشعار في ازدياد مستمر وكذلك الحاجة إلى آلية فعالة للأمن. أن هذه الشبكات تتعامل مع بيانات في كثير من الأحيان حساسة، وهي تعمل في بيئات غير محمية، حيث يعتبر مفهوم الأمن ضروري. ومع ذلك، بسبب محدودية الموارد وضعف قدرة الحوسبة وموارد الطاقة المحدودة وارتفاع معدلات استهلاك الطاقة في مجال الاتصالات، فإن اختيار نموذج آمن لنشر شبكات الاستشعار يعتبر تحديات حقيقاً.

إن تقنية تجميع البيانات هو أسلوب يستخدم على نطاق واسع لأنه يقلل من عدد الرسائل المرسلة في الشبكة، وبالتالي الحد من استهلاك الطاقة وإطالة مدة عمل الشبكة.

في هذه الأطروحة، ركزنا على قضايا السلامة المرتبطة بتقنية تجميع البيانات في شبكات الاستشعار اللاسلكية. درسنا طريقتين: النهاية إلى النهاية وقفزة إلى قفزة وبينا مزايا وعيوب كل طريقة.

الكلمات المفتاحية: شبكات أجهزة الاستشعار اللاسلكية، أجهزة الاستشعار، تجميع البيانات ، الأمن، الطاقة، النهاية إلى النهاية، قفزة إلى قفزة، كونتيكي.