République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE Dr. TAHAR MOULAY SAIDA

FACULTE : TECHNOLOGIE

DEPARTEMENT : INFORMATIQUE

MEMOIRE DE MASTER

OPTION : Sécurité Informatique et Cryptographie

Thème

Analyse des Traces pour la Détection de l'Attaque Sinkhole

dans le Protocole RPL

Présenté par :
M^elle^ BERRAHOU Amina
M^elle^ KADARI Ouarda

Encadré par :
Mr : HACHEMI Fakhr-eddine

Promotion : Juin 2018

Table des matières

Remerciements

We are very honored to have you as framer sir HACHEMI Fakhr-eddine to our thesis.

We thank you for the kindness and spontaneity with which you have kindly directed this work.

We had the great pleasure of working under your direction, and have found with you the counselor and the guide who received us in all circumstance with sympathy, smile and kindness.

Your indisputable professional competence as well as your human qualities are worth to you the admiration and the respect of all. You are and you will be for us the example of rigor and righteousness in the exercise of the profession.

Please, sir, find in this modest work the expression of our highest consideration, our sincere gratitude and our deep respect.

Our thanks also go to the members of the jury for the interest they have shown in our research by agreeing to review our work and enrich it by their proposals.

Our deepest thanks also go to all the professors.

A special thanks to our friend abir, We are very grateful for the spontaneity and The kindness with which you agreed to help us in this work. Please find, the testimony of our great Recognition and our deep respect.

Résumé

Titre : Analyse des Traces pour la Détection de l'Attaque Sinkhole dans le Protocole RPL.

Dans l'Internet des Objets le protocole RPL est un des principales protocoles de routage. La nature des objets et les contraintes qui posent, sont décisif pour le choix de la méthode de détection des attaques. Compte tenu de la nature des réseaux RPL, il est obligatoire d'identifier et d'analyser les attaques auxquelles ce protocole est confronté. L'attaque sinkhole est considérée parmi les plus désastreuse.

Dans ce travail, nous proposons une approche, pour la détection de l'attaque sinkhole, se basant sur les graphiques de contrôle. Nous exploitons les fichiers de traces pour calculer le nombre de message de contrôle délivré par chaque nœud. Ces données, dans le cas où l'attaque n'est pas activée, seront exploitées pour déterminer la Limite Supérieur de Contrôle (LSC). Dans le cas où l'attaque est activée, tous les nœuds qui seront au dessus du LSC, dans le graphique de contrôle, seront considérés comme nœuds affectés par cette attaque.

Mots clés : IoT, Sinkhole, Shewhart, Graphiques de contrôle, RPL.

Abstract

Title : Trace Analysis for Sinkhole Attack Detection in the RPL Protocol.

In the Internet of Things the RPL protocol is one of the main routing protocols. The nature of the objects and the constraints that arise, are decisive for the choice of the method of detection of the attacks. according to the nature of RPL networks, it is obligation to identify and analyze the attacks that this protocol is facing. The sinkhole attack is considered among the most disastrous. In this work, we propose an approach, for the detection of the sinkhole attack, based on the control graphs. We use the trace files to calculate the number of control messages delivered by each node. This data, in case the attack is not activated, will be exploited to determine the Upper Control Limit (LSC). In case the attack is activated, all the nodes that will be above the LSC, in the control graph, will be considered as nodes affected by this attack.

Keywords : IoT, Sinkhole, Shewhart, Control charts, RPL.

Introduction générale

L'Internet des Objets ou IoT (Internet of Things) est une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution. En réalité, l'Internet des Objets n'est pas figée. Elle recoupe des dimensions d'ordres conceptuel et technique. Les objets connectés produisent de grandes quantités de données dont le stockage et le traitement entrent dans le cadre de ce que l'on appelle les big data. Il peut s'agir de capteurs qui servent à la tracabilité des biens pour la gestion des stocks et les acheminements. Dans le domaine de l'environnement, il est question de capteurs surveillant la qualité de l'air, la température, le niveau sonore, l'état d'un bâtiment etc. Les objets d'IoT sont connectés entre eux grâce à des protocoles de routages différents des protocoles utilisés sur les réseaux télécoms ou informatiques.

RPL (The IPv6 Routing protocol for low power and lossy networks) est un protocole de routage proactif à vecteur de distance qui construit un DODAG (Destination Oriented DirecteAcyclic Graph) pour l'acheminement des données vers la station de base. Le DODAG construit permet à chaque nœud de transmettre les données qu'il a récolté jusqu'au DODAG root (racine). Chaque nœud sélectionne un parent selon une métrique de routage donnée et une fonction objective. Les données récoltées sont acheminées de fils à parent jusqu'à la racine, ces métriques sont les informations qui seront prises en compte pour la création de la topologie.

Le protocole RPL est soumit à plusieurs attaques, celle qui visent les ressource (attaques directes, attaques indirectes), d'autres la toplogie (sub-obtimisation, isolation), et celles qui vise le trafic (Eavesdropping, Misappropriation) que nous allons les voir en détail dans ce travail. Notre recherche s'est focalisée sur l'attaque par toplogie, plus précisément l'attaque sinkhole. Il suffit, de placer un nœud malicieux sur un chemin de

communication, il deviendra un nœud routeur avec des critère de performances idéales (le plus court chemin, réserves d'énergie assez conséquente) pour attirer vers lui le trafic réseau. Les paquets de données ainsi que ceux de contrôles sont routés, créant ainsi une fracture du chemin passant par ce nœud (Sinkhole).

En se basant sur l'étude statistique, par la méthode des graphiques de contrôle (Control Chart) de Walter A. Shewhart, nous proposons une approche pour montrer l'impact de l'attaque sinkhole sur un réseau RPL. Notre approche, permet en calculant statistiquement, le nombre de messages de contrôle DIO (DODAG Information Object) et DAO (DODAG Destination Advertisement Object) envoyé par chaque nœud, de déterminer les nœuds affecter par un seul nœud attaquant sinkhole. Nous réalisons une série de simulations dans le simulateur CONTIKI/COOJA, en variant les topologies,le nombre de nœuds, la disposition du nœud attaquant et dans le cas d'une attaque activée ou non. Nous exploitant le données d'une simulation sans attaque pour calculer le seuil (Limite Supérieure de Contrôle : LSC), par la suite les Graphiques de contrôles seront exploités pour déterminer les nœuds qui dépassent le seuil.

Le présent mémoire sera structuré comme suit :

- **Le premier chapitre** présente le contexte général, une présentation du concept d'Internet des Objet ou IoT (Internet of Things) ainsi que le protocole de routage dans es IoT à savoir RPL (Routing Protocol for Low power and lossy networks).
- **Le deuxième chapitre** nous présentons une taxonomie des attaques dans les réseaux RPL.
- **Le troisième chapitre** nous présentons notre approche pour la détection de l'attaque sinkhole en se basant sur la méthode de Shewhart. Nous réalisons une suite de simulation, suivi par une analyse détaillé pour comprendre l'impact de cette attaque.
- **Une conclusion** nous concluons ce travail par des perspectives.

Chapitre

1 Le Protocole RPL et l'Internet des Objets (IoT)

Sommaire

1.1. Introduction

L'Internet des objets (IoT) est une infrastructure, qui permet d'interconnecter différents types d'objets intelligents, autre que les ordinateurs et les téléphones mobiles, pour une qualité de service améliorée dans différents domaines d'application. Les réseaux de capteurs sans fil (RCSFs) comme une composante vitale de l'IOT, permettent la représentation des caractéristiques dynamiques du monde réel dans le monde virtuel de l'Internet. Ainsi, le standard IPv6 (Internet Protocol version 6) s'est étendu en une version compressée (6LoWPAN : IPv6 over Low power Wireless Personal Area Networks) l'IOT recouvre tous les appareils électroménagers communicants, les capteurs (thermostat, détecteurs de fumée, de présence...), les compteurs intelligents et systèmes de sécurité connectés des appareils de type box domotique, il est également très visible dans le domaine de la santé et du bien-être avec le développement des montres connectées, des bracelets connectés et d'autres capteurs surveillant des constantes vitales.

1.2. Typologie des objets

Avec l'evènement de l'Internet des objets, tout objet peut être connecté n'importe quand et n'importe où. il existe différent types d'objets, qu'on appel des objets intelligents, capables de récolter des informations, environnementales ou comportementales (état de l'objet lui-même ou des objets contextuels), de les traiter et de les communiquer sur Internet.

CisCo prévoit que d'ici quelques années, spécifiquement en 2020, l'Internet des objets sera une réalité et le nombre d'objets connectés dépassera les 50 milliards[10].

On distingue différents types de dispositifs connectés à l'IoT, ou ceux qui permettent de connecter d'autres objets à Internet, dont on cite principalement :

1.2.1. Les objets d'identification

Codes barre, marqueurs RFID et autres dispositifs miniaturisés qui servent à l'identification et la traçabilité des objets sur lesquels ils sont collé.

1.2.2. Les capteurs

Les capteurs dans l'IoT permettent de récolter des informations contextuelles concernant les objets dans lesquels ils sont intégrés, ou les environnements sur lesquels ils sont déployés.

1.2.3. Les drones

Un drone désigne un aéronef miniature sans pilote, pouvant porter des charges utiles, communiquer et exécuter des commandes en toute flexibilité. Les drones sont utilisés dans des applications civiles aussi bien que dans des applications militaires.

1.2.4. Smartphones et tablettes électroniques

Les smartphones et les tablettes qui sont déjà connectés à Internet par le biais de diverses technologies (Wi-Fi, 3G, 4G) permettent aux utilisateurs de communiquer à distances avec les autres types d'objets connectés dans l'IoT. Il est possible de les utilisés pour exécuter des taches à distance en supervisant ou ordonnant d'autres objets connectés (smarthouse).

1.3. Les Réseaux de Capteurs Sans Fil (RCSF)

Les Réseaux de Capteurs Sans Fil (RCSF) ou Wireless Sensor Networks (WSN) en anglais sont considérés comme un type particulier de réseaux ad hoc. Les nœuds de ce type de réseaux consistent en un grand nombre de capteurs capables de s'auto organiser, de récolter et de transmettre des données environnementales ou comportementales d'une manière autonome.(figure 1.1).

1.3.1. Essor des réseaux de capteur

Les progrès techniques et technologiques réalisés ces dernières années, ont permis de créer des objets communicants, appelés "nœuds capteurs" ou couramment "capteurs" de plus en plus petits et performants et avec une bonne autonomie énergétique. Ces capteurs sont équipés d'une unité de captage (de mesure), d'une unité de calcul, d'une unité

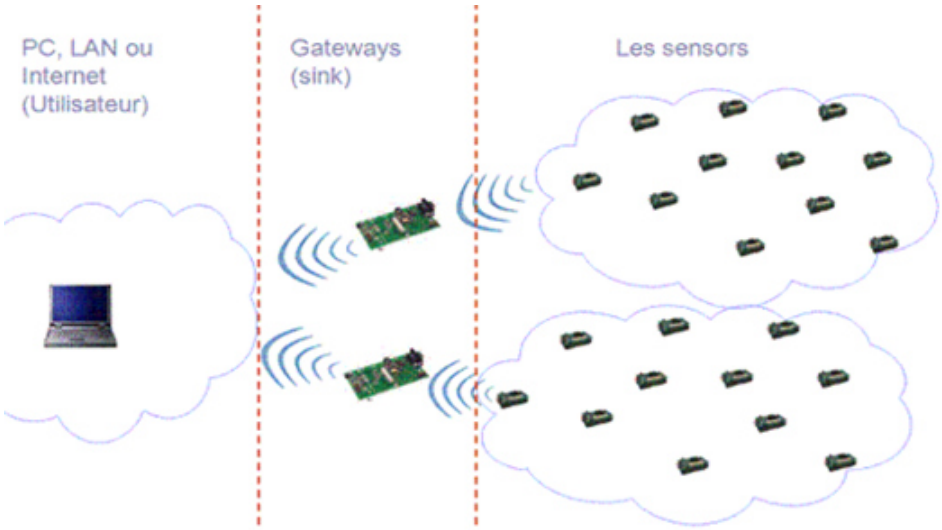

Figure 1.1 – *Architecture d'un RCSF*

de mémorisation (mémoire) et d'une unité de communication (radio). Pour s'alimenter d'énergie, ces nœuds capteurs sont munis de batteries ou d'un système de récupération d'énergie à partir de l'environnement.

En vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, des nœuds capteurs sont déployés sur une zone de surveillance pour former un réseau de capteurs sans fil (RCSF) ou WSN. Un point de collecte est appelé puits (ou sink en anglais), c'est un nœud particulier doté d'une puissance de calcul supérieure et d'une quantité d'énergie potentiellement infinie. Ce puits peut être connecté à Internet ou possède un lien radio de type GSM ou GPRS qui lui permet d'envoyer des données ou des alertes à un centre de contrôle pour l'utilisateur final.

1.3.2. Réseaux sans fil

Un réseau sans fil (wireless network en anglais) est un réseau informatique qui connecte différents hôtes ou nœuds par des ondes radios. Ils constituent avant tout une alternative aux réseaux câblés. Leur compatibilité avec les réseaux câblés permet également de les ajouter comme extension. C'est une technique qui permet aux particuliers, aux réseaux de télécommunications et aux entreprises de limiter l'utilisation de câbles entre diverses localisations.

1.3.3. Définition d'un nœud capteur

Un capteur est un petit dispositif électronique équipé d'une source d'énergie limitée. Les capteurs peuvent être placés ou semés dans une zone d'intérêt pour la surveiller, formant ainsi un réseau de capteurs. Les dispositifs de détection sont conçus pour être capables de former un réseau sans fil autonome, pouvant détecter des données et de les délivrer à un ensemble spécifié de destinations. Les nœuds peuvent détecter les changements environnementaux (température, humidité, pression, etc.) et les signaler à un centre de contrôle via une station de base. Les nœuds capteurs se voient très utiles pour un déploiement dans des environnements hostiles ou dans des grandes zones géographiques. La surveillance des zones est une application courante des réseaux de capteurs (figure 1.2).

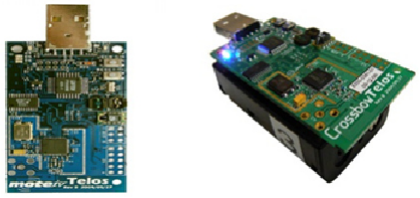

Figure 1.2 – *Mote telosb*

1.3.4. Définition d'un réseau de capteurs sans fil

Un Réseau de Capteurs Sans Fil (RCSF) ou Wireless Sensor Network (WSN) est un réseau informatique composé de petits dispositifs autonomes, fixés ou dispersés aléatoirement dans une zone d'intérêt, utilisant des capteurs coopérant pour surveiller des conditions environnementales ou physiques, comme la température, le son, les vibrations, la pression, le mouvement, etc. Puisque les réseaux de capteurs sans fil peuvent être déployés dans des terrains inaccessibles, la position des nœuds capteurs ne peut être prédéterminée. En conséquence, un système de localisation est requis afin de fournir les informations de position aux nœuds. Parmi les domaines d'application on trouve la santé, le domaine militaire, et de la sécurité.

1.3.5. Caractéristiques des RCSFs [2]

1. Un grand nombre de nœuds
   Scalabilité (zigbee +65 000 sensors).

2. Accès sans fil
   Les interférences sont inévitables (Liens radios perturbés dans un hôpital)
   Ressources limitées.

3. Calcul (4MHz), énergie (Piles AAA), mémoire(512-1MB)
   Gestion d'énergie.

4. Alimentation par batterie.

5. Personne n'ira changer les batteries.

6. Différents modes de veilles
   Idle Mode – 6 mW
   CPU OFF, all peripherals ON
   CPU "woken up" by interrupts
   CPU and most peripherals OFF
   External Interrupts, 2 Wire Interface, Watchdog ON Mode de déploiement

7. Déploiement dans la nature
   Présence d'intrus menant des attaques de sécurité
   Capture des noeuds
   Posés à un endroit précis
   Topologie pré-configurée
   Dispersés aléatoirement

8. Algorithme d'auto-organisation.

1.3.6. Systèmes d'exploitation pour capteurs

Un système d'exploitation pour les nœuds capteurs dans un RCSF est un système adapté aux exigences technologiques et des caractéristiques du nœud capteur lui même, tel que le faible espace mémoire, la capacité de calcul impuissante et la contrainte sérieuse d'énergie. Parmi les systèmes d'exploitation les plus répendus.

1.3.6.1. TinyOS [1]

Le système d'exploitation TinyOS (tiny operating system) est l'un des premiers systèmes d'exploitation conçus pour être adapté aux réseaux de capteurs sans fil. TinyOS est un système open source développé par l'université américaine de Berkeley, écrit en langage NesC (langage syntaxiquement proche du C) qui est orienté composants. TinyOS est dirigé par les évènements (event-driven), c'est-à-dire que les traitements ne s'effectuent que lors d'un stimulus découlant de leurs environnement, en outre ce dernier est capable d'insérer rapidement les changements et les nouveautés liés à l'application, et/ou à la topologie du réseau. Il est réputé par sa bibliothèque particulièrement complète car elle comprend les protocoles réseaux, des pilotes de capteurs ainsi que des outils d'acquisition de données. Ces derniers peuvent être utilisés directement ou adapter à une application précise.

1.3.6.2. Contiki [1]

Contiki est un système d'exploitation écrit en langage C, portable et open source pour capteurs miniatures. Contiki est spécialement conçu pour respecter les contraintes des RCSFs, en particulier, celles qui sont liées aux limitations de l'espace mémoire (il en occupe environ 32 kilo-octets de ROM et 4 kilo-octets de RAM).

Contiki contient un noyau événementiel, au dessus duquel les programmes d'application peuvent être chargés dynamiquement et déchargés au moment de l'exécution. Les processus Contiki utilisent le protothreading ; un style de programmation qui présente un bon compromis entre la programmation évènementielle et la programmation par multithreading. En plus de protothreading, Contiki supporte également la préemption entre les threads. Pour la communication, Contiki implémente deux mécanismes : Rime et uIP. Le premier mécanisme consiste en une couche située juste au-dessous des applications. Telle couche fournit un ensemble d'instructions de communication. Quant à lui, le deuxième mécanisme (uIP : micro IP) est une implémentation adaptée d'une pile protocolaire basée IP (les protocoles : TCP (Transmission Control Protocol), UDP (User Datagram Protocol), IP (Internet Protocol), ICMP (Internet Control Message Protocol)). L'adoption de tel mécanisme de communication rend possible la communication directe entre un capteur et n'importe quel hôte IP. De plus, Contiki offre d'autres fonctionnalités comme un serveur telnet, un serveur web et trois environnements de simulation : Cooja, MSPsim, Netsim.

1.3.7. Les technologies de transmission dans les RCSFs

La communication sans fil dans les réseaux de capteurs est extrêmement importante et critique. Les RCSFs peuvent en supporter plusieurs types dont l'efficacité des communications et la conformité aux particularités du réseau et/ou l'application figée sont des critères clés pour le choix de telle ou telle technologie de transmission sans fil.

1.3.7.1. Wi-Fi (IEEE 802.11a/b/g/n)

La technologie Wi-Fi (IEEE 802.11) permet la connexion d'un réseau local sans fil. Elle est disponible en plusieurs types : A, B, G et N. et récemment les normes AC et AD. La différence entre ces types tourne essentiellement autour du débit maximal qu'un dispositif connecté puisse atteindre et la portée. Wi-Fi utilise la bande de fréquence ISM (Industrial, Scientific and Medical) 2.4 GHz ou la bande 5GHz. Cette technologie est caractérisée par un débit théorique nettement élevé allant de 11Mb/s (pour IEEE 802.11b) jusqu'à 54Mb/s (pour IEEE 802.11a,g). L'avantage du Wi-Fi est qu'il est couramment utilisé et donc, les nœuds capteurs peuvent être facilement connectés aux réseaux WLAN (Wireless Local Networks) existants. A côté de ces avantages, cette technologie est inappropriée pour les réseaux de capteurs standards, en raison de la forte consommation d'énergie induite, ainsi que la complexité de sa pile protocolaire.

1.3.7.2. Bluetooth (IEEE 802.15.1)

La technologie Bluetooth a été initiée en 1994 et actuellement gérée par le groupe SIG (Special Interest Group), elle a été standardisée sous la norme IEEE 802.15.1. Bluetooth est conçu pour fonctionner sur des appareils à faible puissance et faible consommation d'énergie, il a comme but la mise en œuvre des réseaux à portée personnelle où le transfert de données se fait par un débit moyen.

1.3.7.3. ZigBee (IEEE 802.15.4)

ZigBee est une association de plusieurs groupes de recherche qui visent le développement d'un standard global, complet et ouvert pour les communications sans fils avec un coût réduit et une basse consommation d'énergie. Il est fondé sur le standard IEEE

802.15.4 qui définit les couches basses (la sous-couche MAC et la couche physique) pour les réseaux WPAN à très faible débits LRWPAN (Low Rate Wireless Personal Area Networks).

1.4. La couche d'adaptation 6LoWPAN [1]

Vu le nombre important des capteurs qui sont déjà connectés à Internet et ceux qui le seront dans les années à venir et qui seront beaucoup plus nombreux, nécessite d'avoir des adresses unique dans l'Internet du futur. Cependant, le plan d'adressage du protocole IPv4 qui est codé uniquement sur 32 bits, et qui est déjà saturé, ne peut pas satisfaire telle exigence. Pour remédier à cette problématique, l'utilisation du protocole IPv6 qui est caractérisé par un espace d'adressage super large (adresses codée sur 128 bits et la possibilité d'adresser 340 sextillions, soit $340x10^{36}$, d'objets) est avérée incontournable. Néanmoins, le protocole IPv6 est assez coûteux en espace mémoire et en énergie nécessaire pour la communication des datagrammes IPv6 de tailles importantes. Le standard 6LoWPAN définit une couche d'adaptation des datagrammes IPv6 pour les réseaux de capteurs connectés.

1.5. Le routage dans les réseaux 6LoWPANs [3]

Le routage dans les réseaux 6LoWPANs est la fonctionnalité vitale qui assure le bon acheminement des datagrammes 6LoWPAN entre les capteurs appartenant au même réseau ou entre le routeur de bordure 6BR et les nœuds capteurs extrêmes. Deux mécanismes sont définis par l'IETF pour prendre en charge le routage dans ce type de réseaux : le routage maillé au niveau de la couche 6LoWPAN et le routage par le protocole RPL (Routing Protocol for Low power and lossy networks) du groupe RoLL.

1.5.1. Le routage maillé

Ce mécanisme exploite les informations de la couche MAC, plus précisément les adresses MAC, pour réaliser le routage des datagrammes IPv6 compressés (et fragmentés) au niveau de la couche d'adaptation 6LoWPAN. La communication entre la source et la

destination est considérée comme un seul saut IP dont les nœuds intermédiaires (les routeurs) prennent la décision de routage en se basant sur l'analyse de l'adresse MAC de la destination. Si celle-ci ne correspond pas à l'adresse MAC d'un nœud de relai, ce dernier se rend compte que le fragment (et le datagramme) ne lui est pas destiné et consulte donc sa table de routage au niveau liaison pour trouver le nœud prochain. Dans ce cas, les fragments sont acheminés indépendamment les uns des autres. Cela veut dire que les fragments du même datagramme peuvent emprunter différents chemin pour arriver à leur destination finale. Le problème avec cette solution est qu'en cas de perte d'au moins un fragment, la perte ne peut être détectée qu'au niveau de la destination finale et dans tel cas, tous les fragments (y compris le manquant) doivent être retransmis à nouveau pour la récupération.

1.5.2. Le protocole RPL

Le protocole RPL est un protocole de routage IPv6 destiné aux réseaux 6LoWPAN dans l'Internet des objets. il forme une topologie dynamique et optimisée avec l'évitement des boucles et la considération des paramètres de qualité de service pour l'acheminement des datagrammes IPv6 depuis et vers les nœuds capteurs. Chaque nœud intermédiaire se comporte comme un routeur IP, il réassemble d'abord tous les fragments pour reconstruire le datagramme IPv6 initial ensuite, il analyse l'adresse IPv6 de destination pour décider si le paquet va passer à la couche transport ou bien s'il doit être communiqué vers un autre nœud capteurs, jusqu'à ce qu'il arrive à la bonne destination finale.

RPL construit un graphe acyclique dit DODAG (Destination Oriented Directed Acyclic Graph) qui route les informations vers ou depuis une seule destination appelée racine DODAG. Dans certains cas, le même réseau physique 6LoWPAN devrait être optimisé pour supporter plusieurs applications ayant chacune son propre graphe (communément appelée instance RPL), construite selon une métrique de routage bien déterminée. La métrique peut être le niveau d'énergie résiduelle des nœuds capteurs dans le réseau 6LoWPAN, le nombre de transmissions nécessaire pour atteindre la racine (EXT), le délai moyen des communications, le taux de pertes, etc. Lors de la construction du graphe, les nœuds utilisent la fonction objective qui définit la méthode de calcul de la métrique du routage, et s'échangent quatre types de messages : DIO (DODAG Information Object), DIS (DODAG Information Solicitation), DAO (DODAG Destination Advertisement Object) et DAO-ACK (DAO Acknowledgement). Le message DIO est diffusé en premier lieu par

le 6BR (la racine) pour déclencher le processus de construction du graphe. Les nœuds capteurs voisins de la racine reçoivent le message et décident s'ils peuvent joindre le graphe ou non (la décision dépend de plusieurs facteurs tels que la fonction objective et le coût du chemin annoncé). Une fois le nœud a rejoint le graphe, il a automatiquement une route vers la racine. Si le nœud est configuré pour être un routeur, il diffuse à son tour, sa connaissance locale sur le graphe (ses liaisons) à ses voisins.

RPL échange des informations associées à un DODAG selon un ensemble de messages de contrôle ICMPv6. Il en existe 3 types :

1.5.2.1. Le DIO

DODAG Information Object C'est un message de contrôle envoyé en multidiffusion (multicast) par un nœud et retransmis par les autres à leurs voisins (toujours en multi-diffusion), il permet à un nœud de découvrir une instance RPL et de la rejoindre grâce aux informations de routes ascendantes.

1.5.2.2. Le DIS

DODAG Information Sollicitation, littéralement sollicitation d'information DODAG, C'est le message envoyé en multidiffusion (multicast) par un nœud lorsqu'il rejoint un réseau pour demander des informations sur le DODAG, en d'autres termes il demande un message DIO.

1.5.2.3. Le DAO

Destination Advertising Object C'est un message envoyé en monodiffusion (unicast) par les nœuds pour propager les informations de destinations vers le haut du DODAG. Ce qui veut dire que chaque nœud envoi a tous ses parents toutes ses routes descendantes connues. Les nœuds mettent donc à jour leur table de routage à chaque réception d'un DAO.

1.6. La Construction du DODAG [1]

La construction du DODAG est basée sur le processus Neighbor Discovery (ND), littéralement Découverte des voisins, qui se résume en deux opérations principales :

a La propagation des messages de contrôle DIO émis par la racine DODAG et qui sont retransmis par les nœuds vers leurs voisins pour construire des routes dans la direction ascendante.

b La propagation de messages de contrôle DAO émis et retransmis par les nœuds clients jusqu'à la racine DODAG, pour construire des routes dans la direction descendante.

Afin de construire un nouveau DODAG, la racine DODAG diffuse donc un message DIO pour annoncer son DODAGID, sa fonction Objectif, ainsi que des informations pour permettre aux nœuds de déterminer leur rang dans le DODAG. Ce message sera reçu par un nœud client qui peut être un nœud disposé à rejoindre le réseau ou un nœud déjà joint. Lorsqu'un nœud est disposé à rejoindre le DODAG et reçoit un message DIO, il :

1. Ajoute l'adresse de l'émetteur DIO à sa liste de parents .
2. Calcule son rang selon la Fonction Objectif, de sorte que le rang du nœud est supérieur à celui de chacun de ses parents,
3. Il transmet ensuite le message DIO avec les informations de rang actualisées. Ensuite le nœud client choisit son parent préféré parmi la liste de ses parents, ce sera donc le nœud par défaut par lequel le trafic est envoyé dans la direction ascendante.

Lorsqu'un nœud déjà associé à un DODAG et reçoit un autre message DIO, il peut procéder de trois manières différentes :

a Rejeter le message DIO selon certains critères spécifiés par RPL.

b Traiter le message DIO pour maintenir son rang dans le DODAG actuel.

c Améliorer son rang dans le DODAG si ce DIO le permet.

En obtenant un rang inférieur, le nœud améliore donc sa position dans le DODAG chaque fois qu'un nœud change de rang, il doit supprimer tous les nœuds de sa liste de parents dont les rangs sont supérieurs ou égaux au nouveau rang calculé pour éviter les boucles de routage.

1.6.1. Les messages DIS

est utilisé par les nœuds pour demander des informations concernant le graphe à partir des nœuds voisins qui vont répondre en envoyant un message DIO.

1.6.2. Les messages de type DAO

sont utiles pour annoncer la présence du nœud à son parent. Ce dernier met à jour sa table de routage en y rajoutant une entrée correspondante au nœud fils. Le processus se reproduit récursivement et d'une manière ascendante jusqu'à ce que l'on arrive à la racine.

1.6.3. Le message DAO-ACK

est envoyé par le nœud parent au nœud fils, en réponse à son message DAO (pour en accuser réception).

De plus, RPL supporte deux techniques de routage :

le routage par source de données (sans état) et le routage avec décision local du chemin (avec état). Dans le routage par source de données, la totalité du chemin à emprunter est mentionnée dans le paquet, et les nœuds intermédiaires le passe jusqu'à sa destination finale en se basant sur ces informations.

En revanche, dans la deuxième technique, le paquet porte uniquement l'adresse de la destination finale, et le routage est décidé au niveau de chaque nœud intermédiaire suivant les informations contenues dans une table de routage locale. La table de routage comporte des informations pour la distinction des flux ascendant (orientés vers la racine) des flux descendants (orientés vers les nœuds capteurs). Le nœud racine maintient donc, une liste complète de tous les nœuds de l'arborescence.

Notons que pour l'évitement des boucles de routage, chaque nœud doit calculer sa position (ou rang) dans la hiérarchie par rapport à la racine. La valeur du rang devient importante plus la distance entre la racine et le nœud est importante. Des considérations relatives à la métrique du routage peuvent affecter la procédure de calcul de la position. Ainsi, les communications locales entre les nœuds capteurs ayant un parent en commun, ne nécessitent pas de passer par la racine. Cependant, les nœuds n'ayant pas une racine secondaire commune doivent passer par la racine principale (le routeur de bord).

1.7. Conclusion

Dans ce chapitre nous avons expliqué le nouveau concept qui est l'Internet des Objets (IoT), et les réseaux de capteurs sans fil qui sont une composante principale. Le protocole RPL est standardisé pour router les données dans ce type de réseau, qui sont contrarié par certains paramètres physiques et techniques.

Le protocole RPL est étudié en détail afin de connaître son principe pour créer un graphe DODAG, et par la suite pouvoir remédier au problème de sécurité qui le cible. Dans le chapitre suivant, nous détaillerons le problème de sécurité dans ce type de réseaux ainsi que les différentes attaques qui le vise.

CHAPITRE

2 Les Attaques dans le Protocole RPL

Sommaire

2.1. Introduction

Avec l'émergence des réseaux de capteurs sans fil, des réseaux a hôtes autonomes et à infrastructure non prédéfinie utilisés dans des domaines très variés tels que la détection de flux de radiation, le suivi d'objets en déplacement et leur positionnement, Les réseaux de capteurs sont composés d'un nombre important de petits appareils opérant de façon autonome et communicant entre eux via des transmissions à courte portée. Les nœuds capteurs sont conçus pour être déployés d'une manière dense dans des endroits hostiles et difficiles d'accès, d'où la nécessité de limiter au maximum leurs dimensions physiques qui s'obtiennent impérativement au détriment des capacités de calcul de traitement et de ressources énergétiques.

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

2.2. Les objectifs de la sécurité dans les réseaux de capteurs [5]

Les solutions de sécurité destinées aux réseaux de capteurs doivent remplir un ou plusieurs services de sécurité :

2.2.1. La confidentialité

Seules les entités autorisées peuvent accéder les données échangées entre les entités communicantes. Les données doivent donc être chiffrées à l'aide des algorithmes de cryptage suffisamment robustes. D'autre part, la confidentialité des programmes des nœuds capteurs doivent être garantie ; le nœud capteur ne doit en aucun cas se permettre la lecture de son contenu par des parties non autorisées. Donc, mêmes les données propriétaires au capteur, comme les clés cryptographiques, le programme du capteur et son identificateur, doivent être protégées.

2.2.2. L'intégrité

Le mécanisme de sécurité doit garantir que les données ne seront pas altérées le long de leur passage vers la station de base. Les deux entités doivent implémenter des techniques de détection de toute modification de données.

2.2.3. L'authentification

Il arrive qu'un attaquant ne cause pas que la modification des paquets qu'il intercepte mais aussi, il peut forger et injecter des paquets falsifiés dans le réseau. Dans tel cas, le nœud capteur doit pouvoir vérifier la validité des identificateurs des nœuds sources de données qui lui parviennent.

2.2.4. La fraîcheur de données

Dans la majorité des applications des réseaux de capteurs, les données récentes sont vivement suggérée. Un attaquant peut violer cette propriété, en rejouant plusieurs fois des anciens messages. De ce fait, les nœuds capteurs et la station de base doivent mettre en place des mécanismes appropriés pour s'assurer de la fraîcheur des données communiquées.

2.2.5. La disponibilité

Les RCSFs sont des réseaux orientés-services, ce qui veut dire que le réseau est spécialement mis en place pour rendre un service bien déterminé et souvent assez critique. Donc, même dans le cas où le réseau de capteurs est ciblé par des attaques, il doit résister tant que possible et préserver la disponibilité de ses ressources et services.

2.2.6. La sécurité de la localisation

Le réseau de capteurs a souvent besoin des informations précises de localisation concernant des objets contrôlés par les capteurs et/ou les capteurs eux-mêmes. Telles informations doivent nécessairement être protégées contre toute interception illégale ou manipulation mal intentionnée.

2.3. LES MODÈLES D'ATTAQUES DANS LES RÉSEAUX DE CAPTEURS

Selon des critères bien spécifiques, comme l'ampleur de l'attaque, la puissance de l'attaquant, l'appartenance ou non de ce dernier au réseau, on distingue différentes classes d'attaques :

2.3.1. Les attaques accidentelles et les attaques intentionnelles

Les attaques accidentelles sont représentées par défaillances que subisse un nœud capteur depuis son entourage. Cependant, les attaques intentionnelles et qui sont les plus fréquentes et les plus nuisibles aux RCSFs, sont gérées par des personnes malveillantes ayant un objectif malicieux.

2.3.2. Les attaques externes et les attaques internes

Les attaques externes proviennent des nœuds qui n'appartiennent pas au réseau de capteurs (les nœuds intrus), et les attaques internes sont exercées par les nœuds de compromission qui font partie du réseau.

2.3.3. Les attaques impuissantes et les attaques puissantes

Dans les attaques impuissantes (mote-class), l'attaquant utilise un certain nombre de nœuds ayant des capacités similaires à celles des nœuds capteurs du réseau pour l'attaquer,les attaques puissantes (laptop-class) sont les plus dangereuses,l'attaquant fait appel à des dispositifs à fortes capacités.

2.3.4. Les attaques passives et les attaques actives

Dans le cas où l'attaquant ne fait qu'écouter et analyser illicitement le trafic qui transite entre les nœuds capteurs, l'attaque est dite passive. Dans le cas échéant où l'attaquant se permet même de modifier, détourner, bloquer ou forger des données dans le réseau, l'attaque est dite active.

2.4. Les niveaux d'attaques dans les réseaux de capteurs [4]

Les attaques qui ciblent les réseaux de capteurs peuvent opérer dans plusieurs niveaux de la pile protocolaire du capteur. A chaque fois les attaquants exploitent les failles de sécurité des protocoles ou des spécificités d'un niveau donné (physique, liaison de données, routage ou transport de données).

2.5. Le niveau physique

La couche physique est très sensible aux attaques qui exploitent l'accessibilité du support de transmission pour intercepter les communications ou pour causer des problèmes plus grave comme, le brouillage que l'attaquant puisse provoquer en envoyant des signaux parasites qui interfèrent avec les fréquences radio qu'utilisent les nœuds capteurs pour la communication. Une deuxième catégorie d'attaques possibles dans la couche physique des RCSFs, est la falsification des nœuds capteurs (node tampering). L'attaquant dans ce cas capture un nœud et extrait son contenu (ses programmes) à partir de la mémoire de ce même nœud capteur. Donc les clés cryptographiques et les autres informations sensibles seront dévoilées. Le nœud capturé pourrait même être corrompu (l'attaquant modifie le programme du nœud capteur en y insérant des codes malicieux) ou bien, remplacé par un nœud de compromission (qui est généralement plus riche en ressources) que l'attaquant puisse superviser.

2.6. Le niveau liaison de données

Les attaques qui se concentrent dans ce niveau provoquent des collisions avec les communications inter-capteurs ou entre capteurs et station de base. Les collisions intensives causent des ruptures de communications dans le réseau qui en résulte une consommation excessive d'énergie résultante des retransmissions répétées des trames corrompues. Comme solutions possibles à ce problème : l'adoption des techniques préventives adaptées, comme les techniques d'évitement de collisions (CSMA/CA : Carrier Sense Multiple Access with Collision Avoidance) et la méthode d'accès au support à base de (MAC IEEE 80.15.4) où une station dite coordinateur du réseau se fait consacrer pour

la gestion des priorités entre les nœuds capteurs afin de synchroniser les communications entre elle et ces derniers.

2.7. Le niveau routage de données

Le routage de données depuis leurs sources jusqu'à la station de base est une fonctionnalité qui est à la fois vitale et critique dans les RCSFs. Ce mécanisme est exposé à un large éventail d'attaques qui peuvent affecter la phase de construction des routes et même la phase d'acheminement des données.

2.8. Protocole de routage RPL [7]

L'Internet des Objets ou Internet of Things(IoT) se traduit par le déploiement de réseaux avec pertes et à faible puissance appelés réseaux LLN. Ces réseaux permettent à de nombreux équipements embarqués comme des capteurs de pouvoir communiquer entre eux. Un protocole de routage appelé RPL a été spécialement conçu par l'IETF pour répondre aux contraintes spécifiques qu'impose ce type de réseaux. Cependant, ce protocole reste exposé à de nombreuses attaques de sécurité. Le protocole RPL est un protocole de routage à vecteur de distance utilisant IPv6, spécialement conçu par l'IETF pour répondre aux besoins des réseaux LLN. Cette section présente le fonctionnement de ce protocole et les mécanismes de protection existants.

2.8.1. Topologie, instance et fonction objectif

Les noeuds RPL s'interconnectent en formant une topologie spécifique appelée DODAG, c'est-à -dire un graphe acyclique orienté dirigé vers une destination qui est la racine du réseau. Un réseau RPL contient au moins une instance RPL qui elle-même se compose d'un ou plusieurs DODAGs. Chaque instance RPL est associée à une fonction objectif (OF) qui permet d'optimiser la topologie en fonction d'un ensemble de contraintes et/ou de métriques comme la préservation de l'énergie, le chemin le plus court ou la qualité des liens. Un nœud peut faire partie d'un seul DODAG par instance, mais peut participer à plusieurs instances simultanément.

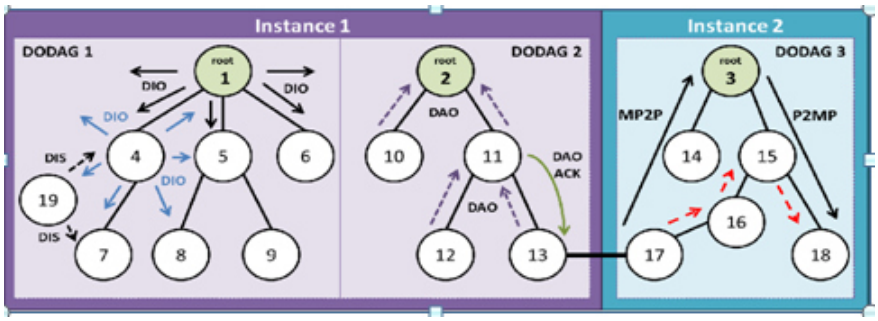

FIGURE 2.1 – *Exemple d'un réseau RPL composé de deux instances et trois DODAG*

2.8.2. MESSAGES DE CONTRÔLE ET CONSTRUCTION DU DODAG

La construction et la maintenance des DODAG sont réalisées grâce à des messages de contrôle ICMPv6 figure 2.2. Plus particulièrement, trois nouveaux messages sont définis : (1) DODAG Information Sollicitation (DIS), (2) DODAG Information Object (DIO) et (3) Destination Avertissement Object (DAO). Un nouveau nœud peut rejoindre un réseau déjà formé en diffusant un message DIS pour solliciter en réponse un message DIO qui contient des informations sur le DODAG comme le numéro de version et l'identifiant du DODAG, l'identifiant de l'instance et l'OF utilisée. Un nœud peut également attendre de recevoir un message DIO diffusé périodiquement par ses voisins. La fréquence d'envoi des messages DIO est déterminée par un temporisateur fondé sur l'algorithme Trickle (appelé également temporisateur Trickle). À la moindre anomalie dans le réseau, le temporisateur Trickle est réinitialisé pour permettre à la topologie de reconverger plus rapidement. Après avoir reçu un message DIO, le nœud calcule son rang en utilisant l'OF spécifiée dans ce message. Le rang d'un nœud correspond à son emplacement dans le graphe par rapport à la racine. La valeur du rang augmente toujours en descendant dans le graphe. C'est donc la racine qui a le rang le plus petit dans le graphe. Si un nœud reçoit des DIO de voisins différents, l'émetteur avec le meilleur rang (le plus petit donc) est choisi comme le parent préféré vers lequel seront envoyés tous les messages à destination de la racine. À la fin de ce processus seulement. les routes ascendantes (i.e. vers la racine) sont construites. Pour établir les routes descendantes, un nœud doit envoyer un message DAO à son parent contenant le préfixe des nœuds situés dans son sous-DODAG. Lorsque le message se propage vers le haut, les préfixes sont agrégés et les routes descendantes deviennent disponibles pour les parents.

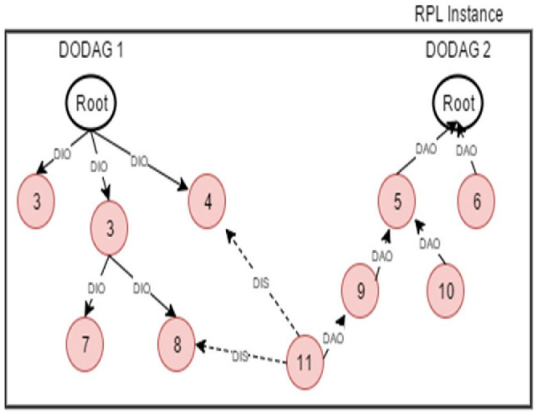

FIGURE 2.2 – *Messages de contrôle et construction du DODAG*

2.8.3. MÉCANISMES DE PROTECTION EXISTANTS [8]

RPL intègre différents mécanismes afin d'éviter les boucles, détecter les incohérences et réparer le graphe. Le rang joue un rôle important pour construire une topologie sans boucle. En effet, un nœud ne peut choisir qu'un parent dont le rang est inférieur au sien, autrement dit tous les nœuds se trouvant dans le sous-DODAG d'un nœud ont un rang supérieur à ce nœud. Si un nœud ne respecte pas cette propriété du rang, le graphe n'est plus acyclique. De plus, pour d'éviter les boucles, si un nœud doit changer son rang, il doit utiliser un mécanisme de poisoning (en annonçant un rang infini) ou de déconnexion (en formant un DODAG temporaire).

Dans les cas où des boucles apparaissent dans le graphe, le protocole RPL fournit une fonctionnalité appelée validation du chemin de données. Des informations de contrôle sont transportées dans les paquets de données via des flags placés dans l'en-tête d'extension IPv6 Hop-By-Hop :

1. Le flag 'O' indique la direction attendue du paquet, i.e., vers le haut ou le bas. Si un nœud place ce flag à 1 le paquet est destiné à un descendant, sinon le paquet est supposé être envoyé à un parent avec un rang inférieur, vers la racine du DODAG.

2. Le flag 'R' indique si une erreur de rang a été détectée par un nœud transférant le paquet. Ce flag est mis à 1 lorsqu'un nœud observe une incohérence entre la direction supposée du paquet indiquée par le flag 'O' et le rang du nœud qui vient de le transférer. Le flag 'R' est utilisé pour réparer ce type d'anomalie appelée incohérence DODAG.

Deux principaux mécanismes de réparation sont utilisés dans les réseaux RPL en cas d'incohérences ou de pannes : la réparation locale et globale. La réparation locale consiste à trouver un chemin alternatif pour router les paquets. Si les réparations locales ne suffisent pas, la racine peut initier une réparation globale en incrémentant le numéro de version du DODAG. Ceci a pour résultat la reconstruction complète du graphe.

2.8.4. Taxonomie des attaques contre le protocole RPL [7]

Les différentes attaques visant le protocole RPL ont été classifiées selon qu'elles menaçaient en priorité les ressources des nœuds, la topologie du réseau et le trafic comme le montre la figure 2.3. Les attaques de la première catégorie ont pour but de consommer l'énergie, la mémoire ou le temps de calcul des nœuds. Les attaques de la seconde catégorie visent la topologie du réseau. Alors que la dernière catégorie concerne les attaques ciblant le trafic.

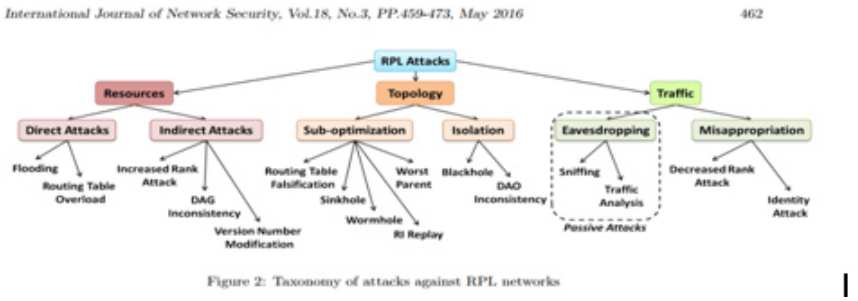

Figure 2.3 – *Taxonomie des attaques contre le protocole RPL*

2.9. Attaques contre les ressources

consistent généralement à faire exécuter par des nœuds légitimes des traitements inutiles épuiser leurs ressources. Cette catégorie d'attaques vise au niveau de l'énergie, de la mémoire ou du traitement du nœud consommateur. Elle peut avoir un impact sur la disponibilité du réseau en encombrant les liaisons disponibles et donc sur la durée de vie du réseau. Nous distinguons deux sous-catégories d'attaques contre des ressources.

2.9.1. ATTAQUES DIRECTES

En cas d'attaques directes, l'attaquant est directement responsable de l'épuisement des ressources. Cela peut généralement être fait en effectuant des attaques ou en exécutant une surcharge attaques par rapport aux tables de routage, lorsque le mode de stockage est actif.

a Attaques d'inondation (flooding) : Les attaques d'inondation consistent à générer une grande quantité de trafic dans un réseau et rendre les nœuds et les liens indisponibles. Dans les réseaux RPL, un attaquant peut soit diffuser des messages DIS à ses nœuds voisins avoir à réinitialiser leur minuterie goutte à goutte, ou, unicast message DIS à un noeud qui doit répondre avec un message DIO. Cette attaque conduit à la congestion du réseau et aussi la saturation des noeuds RPL.

b Attaques de surcharge de la table de routage (routing table overload) : Il est également possible d'effectuer des attaques directes contre des ressources en surchargeant la tables de routage RPL. Le protocole RPL est un protocole proactif. Cela signifie que les noeuds de routeur construisent et maintiennent des tables de routage quand le mode de stockage est activé pour ces nœuds. Le principe de surcharge de table de routage est d'annoncer de fausses routes en utilisant les messages DAO qui saturent la table de routage du noeud ciblé. Cette saturation empêche la construction de nouvelles routes légitimes. Cela peut avoir des impacts sur fonctionnement du réseau et peut entraîner un dépassement de mémoire.

2.9.2. ATTAQUES INDIRECTES

où les attaquants feront que d'autres nœuds génèrent une grande quantité de trafic. Par exemple, une telle attaque peut être réalisée en construisant des boucles dans le réseau RPL de sorte que d'autres nœuds produisent des frais généraux de trafic.

Les attaques indirectes correspondent à des attaques où le nœud malveillant fait que les autres nœuds génèrent une surcharge pour le réseau. Il comprend : attaques d'augmentation du rang, attaques d'incohérence de DAG et attaques par modification du numéro de version :

a Les Attaques d'augmentation du rang (increasing rang attack) : consiste à augmenter volontairement le rang du Nœud RPL afin de générer des boucles dans le réseau. Comme mentionné précédemment, le rang de nœud est toujours croissant vers le bas afin de préserver la structure acyclique du DODAG. Lorsqu'un nœud détermine sa valeur de rang, celui-ci doit être supérieur aux valeurs de rang de ses parents.

Si un nœud veut changer sa valeur de rang, pour mettre à jour sa liste de parents en supprimant les nœuds ayant un rang plus élevé que sa nouvelle valeur de classement. Une fois qu'un nœud a établi l'ensemble des parents dans un DODAG, il sélectionne son parent préféré de cette liste afin d'optimiser le coût de routage lors de la transmission du paquet au nœud racine. Un noeud malveillant annonce une valeur de rang plus élevée que celui qu'il est censé avoir. Pour atténuer cette attaque, le nombre de fois qu'un nœud RPL augmente son rang dans le graphe DODAG doit être surveillée pour déterminer si un nœud peut être considéré comme malveillant ou mal compris. Il est important de remarquer qu'un nœud peut légitimement augmenter sa valeur de rang si elle ne correspond plus à la fonction objectif.

b Attaques d'incohérence de DAG (DAG inconsistency) : Un noeud RPL détecte une incohérence DAG lorsqu'il reçoit un paquet avec un 'O' Down bit défini à partir d'un nœud avec un rang plus élevé et vice-versa. quand la direction du paquet ne correspond pas à la relation de rang. Cela peut être le résultat d'une boucle dans le graphe. Le bit "R" Rank-Error est utilisé pour contrôler ce problème. Quand une incohérence est détectée par un nœud, deux scénarios sont possibles : (i) si l'erreur de classement n'est pas défini, le nœud le définit et le paquet est transféré. Une seule incohérence le long de la trajectoire n'est pas considéré comme une situation critique pour le réseau RPL, (ii) si le bit 'R' est déjà défini, le nœud rejette le paquet et la minuterie de maintien est réinitialisée. En conséquence, les messages de contrôle sont envoyés plus fréquemment. Un nœud malveillant a juste pour modifier les flags ou ajouter de nouveaux flags à l'en-tête. Le résultat immédiat de cette attaque consiste à forcer la réinitialisation du minuteur de diffusion DIO du nœud ciblé. Dans ce cas, ce nœud commence à transmettre des messages DIO plus fréquemment en produisant localement consomme également la batterie des nœuds et impacts sur la disponibilité des liens. Tout le voisinage de l'attaquant est concerné par l'attaque.

c Les attaques par modification du numéro de version (version number modification) : Le numéro de version est un champ important de chaque message DIO. Il est propagé inchangé sur le graphe DODAG et est incrémenté par la racine seulement, chaque fois La reconstruction du DODAG est nécessaire, ce que l'on appelle aussi la réparation globale. Une valeur plus ancienne indique que le nœud n'a pas migré vers le nouveau graphe DODAG et ne peut pas être utilisé comme nœud parent. Un attaquant peut modifier le numéro de version de manière illégitime augmenter ce champ dans les messages DIO quand il les envoie à ses voisins. Tel attaque provoque une reconstruction inutile de l'ensemble du graphe DODAG ressources de nœud.

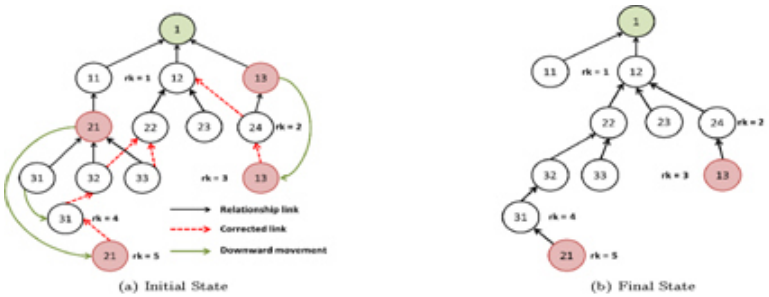

FIGURE 2.4 – *Attaque d'augmentation du rang [7]*

2.10. ATTAQUES SUR LA TOPOLOGIE

Les attaques contre le protocole RPL peuvent également cibler la topologie du réseau. Nous distinguons deux catégories principales parmi ces attaques : la sous-optimisation et l'isolation.

2.10.1. ATTAQUES DE SOUS-OPTIMISATION

En cas d'attaques de sous-optimisation, le réseau ne convergera pas vers la Forme optimale (c'est-à-dire des chemins optimaux) induisant des performances médiocres.

2.10.1.1. ATTAQUE DE FALSIFICATION DE TABLE DE ROUTAGE (ROUTING TABLE FALSIFICATION)

Dans un protocole de routage, il est possible de forger ou de modifier des informations de routage pour faire de la publicité routes falsifier vers d'autres nœuds. Cette attaque peut être effectuée dans le réseau RPL par modification ou forger des messages de contrôle DAO afin de créer de fausses routes descendantes. Cela ne peut être fait que lorsque le mode de stockage est activé. Par exemple, un nœud malveillant affiche des routes vers des nœuds qui ne sont pas dans son sous-DODAG. Nœuds ciblés ont alors des routes erronées dans leur table de routage provoquant une sous-optimisation du réseau. En conséquence, le chemin peut être plus long à induire un retard, des pertes de paquets ou une congestion du réseau.

2.10.1.2. Attaque de puit (sinkhole)

Une attaque alternative consiste à construire un gouffre. Une telle attaque a lieu dans deux étapes. Tout d'abord, le nœud malveillant parvient à attirer beaucoup de trafic par la publicité donnée d'information falsifiées (par exemple, liens ascendants et descendants de qualité supérieure). Puis, après avoir reçu le trafic d'une manière illégitime, l'attaque peut être facilement réalisée grâce à la manipulation de la valeur du rang. En raison de cette publicité falsifiée, le nœud malveillant est plus fréquemment choisi comme parent préféré par les autres nœuds, alors qu'il ne fournit pas de meilleures performances. Ainsi, les routes ne sont pas optimisées pour le réseau. L'attaque modifie la topologie et dégrade les performances du réseau. si l'attaquant décide de laisser tomber tout le trafic, il effectue également attaque du trou noir.

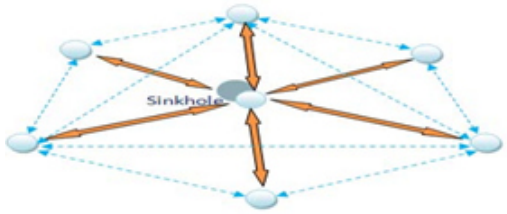

Figure 2.5 – *attaque de puit(sinkhole)*

2.10.1.3. Attaque de trou de ver (Wormhole)

Les attaques de trou de ver sont définies comme l'utilisation d'une paire de nœuds d'attaquant RPL, les nœuds A et B, lié via une connexion réseau privé. Un exemple est illustré à la figure 2.6. Dans ce scénario, chaque paquet reçu par le nœud 13 est transmis à travers le trou de ver au nœud 21 pour être rejoué plus tard. Puisque les rôles sont interchangeables, le nœud 21 peut effectuer les mêmes opérations que le nœud 13. Dans le cas des réseaux sans fil, il est plus facile d'effectuer cette attaque parce que l'attaquant peut envoyer à travers le trou de ver le trafic adressé à lui-même ainsi que tous les trafic interceptés dans la transmission sans fil. L'attaque de trou de ver déforme le chemin de routage et est particulièrement problématique pour les réseaux RPL. Si un attaquant tunnels achemine des informations vers une autre partie du réseau, les nœuds qui sont réellement distants, se voient comme s'ils étaient le même quartier. En conséquence, ils peuvent créer des routes non optimisées à la fonction objective.

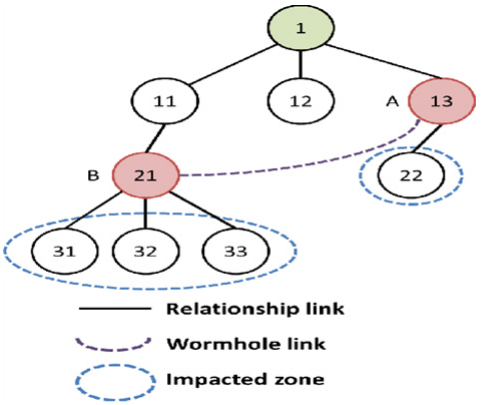

FIGURE 2.6 – *attaque trou de ver (wormhole)*

2.10.1.4. LES ATTAQUES DE RÉPÉTITION D'INFORMATION DE ROUTAGE(ROUTING INFORMATION REPLAY ATTACKS)

Un nœud RPL peut également effectuer des attaques de répétition d'informations de routage. Il enregistre valide contrôler les messages d'autres nœuds et les transmettre plus tard dans le réseau. Au cas où des réseaux dynamiques, cette attaque est assez dommageable parce que la topologie et les chemins de routage sont souvent modifiés. Les attaques de relecture obligent les nœuds à mettre à jour leur table de routage avec des données périmées entraînant une fausse topologie. Le protocole RPL utilise certains compteurs de séquence pour assurer la fraîcheur de l'information de routage tels que le numéro de version des messages DIO ou le numéro de séquence de chemin présent dans l'option d'information de transit des messages DAO .

2.10.1.5. WORST PARENT ATTACKS

Cette attaque appelée "Attaque de rang" consiste à choisir systématiquement le parent le plus défavorisé en fonction de la fonction objectif. Le résultat est que la trajectoire résultante n'est pas optimisée, ce qui induit de mauvaises performances. Cette attaque ne peut pas être facilement abordée parce que le nœud enfants comptent sur leur parent pour acheminer les paquets et cette attaque ne peuvent pas être surveillés par les voisins. Cependant, en utilisant une solution qui reconstruit une vue globale du graphe en fonction de l'information des nœuds devrait détecter cette attaque.

2.10.2. LES ATTAQUES D'ISOLATION(ISOLATION ATTACKS)

Les attaques contre la topologie peuvent également isoler un nœud ou un sous-ensemble de nœuds dans le Réseau RPL ce qui signifie que ces nœuds ne sont plus en mesure de communiquer avec leurs parents ou avec la racine.

2.10.2.1. L'ATTAQUE DE TROU NOIR(BLACKHOLE)

Dans une attaque blackhole, un intrus malveillant supprime tous les paquets qu'il est supposé à transmettre. Cette attaque peut être très dommageable lorsqu'elle est combinée avec une attaque de gouffre causant la perte d'une grande partie du trafic. Cela peut être vu comme un type d'attaque de déni de service. Si l'attaquant est situé à une position stratégique dans le graphique, il peut isoler plusieurs nœuds du réseau. Il y a aussi une variante de cette attaque appelée trou gris (ou attaque d'expédition sélective) où l'attaquant se défait seulement d'un spécifique, des attaques blackhole dans les réseaux RPL à travers un ensemble de simulations Cooja. Ils mis en évidence différents indicateurs pour détecter ces attaques telles que le taux et la fréquence de Messages DIO, taux de livraison des paquets, pourcentage de perte et délai.

2.10.2.2. LES ATTAQUES D'INCOHÉRENCE DAO (DAO INCONSISTENCY ATTACKS)

Les incohérences DAO se produisent lorsqu'un nœud a une route vers le bas qui était précédemment Appris à partir d'un message DAO, mais cette route n'est plus valide dans la table de routage du nœud enfant. RPL fournit un mécanisme pour réparer cette incohérence, appelé récupération de boucle d'incohérence DAO dans la validation du chemin de données. Ce mécanisme permet aux nœuds du routeur RPL d'éliminer les routes descendantes non souhaiter en utilisant le flags Forwarding-Error dans les paquets de données qui indique qu'un paquet ne peut pas être délivré par un nœud enfant. Le paquet avec le 'flag' est renvoyé au nœud parent afin d'utiliser un autre nœud voisin (voir figure 2.7).

2.11. LES ATTAQUES SUR LE TRAFIC

Cette troisième catégorie concerne les attaques ciblant le réseau trafic RPL. C'est principalement comprend des attaques d'écoute d'une part, et des attaques de détourne-

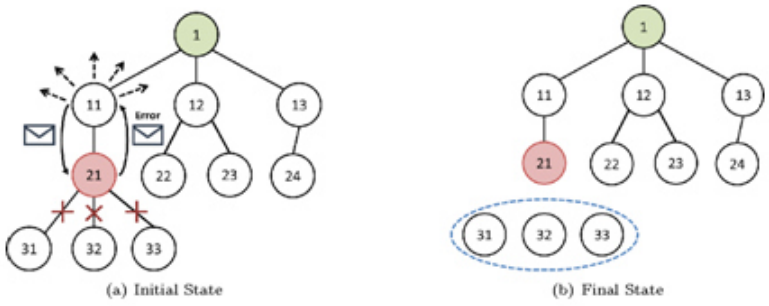

FIGURE 2.7 – *Illustration de l'attaque DAO inconsistency*

ment d'autre part.

2.11.1. LES ATTAQUES DE TENTATIVE D'ÉCOUTE(EAVESDROPPING ATTACKS)

La nature omni présente des réseaux RPL peut faciliter le déploiement de programmes malveillants nœuds effectuant des activités d'écoute comme sniffing et l'analyse du trafic du réseau.

2.11.1.1. SNIFFING ATTACKS

Une attaque sniffing consiste à écouter les paquets transmis sur le réseau. Cette attaque est très fréquente dans les réseaux filaires et sans fil et compromet la confidentialité des communications. Un attaquant peut effectuer cette attaque en utilisant un périphérique compromis ou capturer directement les paquets du support partagé dans le cas des réseaux sans fil. Les informations obtenues à partir des paquets sniffed peuvent inclure topologie partielle, informations de routage et contenu de données. Dans les réseaux RPL, si les messages de contrôle de l'attaquant sniffing il peut accéder aux informations concernant le DODAG configuration telle que DODAG ID, numéro de version, rangs des nœuds situés dans le quartier. En sniffing paquets de données, les attaques peuvent non seulement découvrir le paquet contenu, mais aussi avoir une vue locale de la topologie dans la zone indiscrète , regarder les adresses source / destination. Cette attaque est difficile à détecter en raison à sa nature passive. La seule façon d'empêcher sniffing est le cryptage des messages lorsque l'attaquant est externe. Même si RFC 6550

mentionne le cryptage du contrôle messages en option, les détails techniques sont omis de la spécification mise en œuvre difficile.

2.11.1.2. Les Attaques d'analyse du trafic (Traffic Analysis Attacks)

L'analyse de Trafic vise à obtenir des informations de routage en utilisant les caractéristiques et modèles du trafic sur un lien. Cette attaque peut être effectuée même si les paquets sont crypté. L'objectif est, comme les attaques furtives, de recueillir des informations sur Réseau RPL tel qu'une vue partielle de la topologie en identifiant les parents / enfants attaques avec les informations recueillies. Les conséquences dépendent du rang de l'attaquant. Si celui-ci est proche du nœud racine, il peut traiter une grande quantité de trafic et peut donc obtenir plus d'informations que lorsque le nœud est situé sur le bord d'un sous-DODAG.

2.11.2. Les Attaques de détournement (Misappropriation Attacks)

Dans les attaques de détournement, l'identité d'un nœud légitime est usurpée ou performance sont surclassés. Ces attaques ne sont pas si dommageables pour le réseau RPL Cependant, ils sont souvent utilisés comme une première étape pour d'autres attaques telles que celles vu dans les deux catégories principales précédentes. Ils permettent à l'attaquant de mieux comprendre le réseau et de sa topologie, pour avoir un meilleur accès ou pour intercepter une grande partie du trafic.

2.11.2.1. Les Attaques du rang diminué (Decreased Rank Attacks)

Dans un graphe DODAG, plus le rang est bas, plus le nœud est proche de la racine et ce nœud doit gérer plus trafic. Quand un nœud malveillant illégitimement annonce une valeur de rang inférieur, il sur-revendique sa performance. En conséquence, de nombreux nœuds légitimes se connectent au graphe DODAG via l'attaquant. Cela résulte en l'attraction d'une grande partie du trafic. Grâce à cette opération, le nœud malveillant est capable d'effectuer d'autres attaques telles les attaques d'écoute. Dans le protocole RPL, un attaquant peut changer de rang par la falsification des messages DIO.

2.11.2.2. Les Attaques d'identité (Identity Attacks)

Les attaques d'identité rassemblent à la fois les attaques spoofing et sybil. également appelée L'attaque de clone ID se produit lorsqu'un nœud malveillant prétend être un nœud existant. Dans les réseaux RPL, le nœud racine joue un rôle clé dans un graphe DODAG. Il construit et maintient la topologie en envoyant des informations de routage. Un attaquant peut écouter le trafic réseau pour identifier le nœud racine. Une fois cette identification effectuée, il peut usurper l'adresse de la racine DODAG et prendre le contrôle du réseau. un nœud malveillant utilise plusieurs entités logiques sur le même nœud physique.

2.12. Conclusion

Compte tenu de la nature des réseaux RPL, il est obligatoire d'identifier et d'analyser les Attaques de sécurité auxquelles ce protocole est confronté. Nous avons exposé, en ce chapitre, une taxonomie classant les attaques contre le protocole RPL dans trois Catégories principales. Les attaques contre les ressources réduisant la durée de vie du réseau génération de faux messages de contrôle ou la construction de boucles. Les attaques contre la topologie font converger le réseau vers une configuration sous-optimale ou elles isolent les nœuds. Enfin, les attaques contre le réseau trafic permettant à un nœud malveillant de capturer et analyser une grande partie du trafic.

Chapitre

3

Approche pour la Détection de l'Attaque Sinkhole

Sommaire

3.1. Introduction

Dans ce chapitre nous allons étudier l'attaque sinkhole. Dans ce type d'attaque, l'intrus essaye d'attirer vers lui le plus de chemins possibles permettant le contrôle de la plus part des données circulant dans le réseau. Pour ce faire, l'attaquant doit apparaître aux autres comme étant très attractif, en présentant des routes optimales.

Dans notre approche, nous nous sommes basés sur l'étude statistique des messages de contrôles à savoir les messages DIS, DIO et DAO. Une séries de simulations ont été réalisée, avec un seul nœud malicieux à chaque fois. Nous exploitants les fichiers logs et le trafic qui se déroule dans le réseau. Nous calculons le nombre de messages DIS/DIO/DAO pour chaque nœud dans le cas des simulations sans attaques pour calculer le seuil de 3 sigma (Shewhart), et tracer le graphique de contrôle dans le cas où il y a une attaque.

3.2. Les graphiques de contrôle [9]

Les graphiques de contrôle de la moyenne et de l'étendue de son inventeur du graphique de contrôle (Control Chart) Walter A. Shewhart. Il a publié en 1931 les principes de la variabilité d'un processus en distinguant la variabilité aléatoire naturelle et la variabilité accidentelle. La variabilité naturelle est issue de causes communes de dispersion ou perturbations normales intégrées dans le processus de fabrication sous contrôle. La variabilité accidentelle est due à des causes spéciales occasionnelles et incontrôlées (matières premières aux caractéristiques fluctuantes) Une analyse plus détaillée des causes des variations permettra d'améliorer ses performances et sa régularité.

Un graphique de contrôle est un outil de visualisation d'un processus dans le temps et de mise en évidence de sa stabilité (surveillance des causes spéciales). qui utilise des limites appelées limite supérieure de contrôle (LSC) et limite inférieure de contrôle (LIC). On prélève des échantillons à intervalles de temps réguliers et on reporte sur le graphique la valeur de la moyenne de l'échantillon.

Dans notre cas nous allons étudier la limite supérieur c'est pour cela nous avons proposer un script qui calcule la moyenne du DAO et DIO pendant la période observée. Le calcul de la limite supérieure de contrôle (LSC) à partir de la moyenne et de l'écart

type des données (nombre de message DIO/DAO envoyé par chaque nœud) suivant la formule 3.1 de Shewhart :

$$LSC = \mu + 3 * \sigma \tag{3.1}$$

où μ est la moyenne des données observées formule 3.2 et σ est l'écart type ref-ecart calculer à partir de la variance formule 3.3 qui mesure de la dispersion d'une liste de valeur autour de sa moyenne. Cette valeur, notée V ou Var caractérise la manière dont les données X (variable aléatoire) sont dispersées en mesurant les écarts entre chaque valeur (de la variable) et la moyenne (ou espérance).

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \tag{3.2}$$

$$Var = \frac{\sum_{i=1}^{n} (x_i - X_i)^2}{n} \tag{3.3}$$

$$\sigma = \sqrt{Var} \tag{3.4}$$

3.3. Détection de l'attaque Sinkhole

Dans une attaque sinkhole, le nœud malveillant tente d'attirer un maximum de trafic. Cette attaque est l'une des plus dangereuse pour les RPL, surtout si elle est exploiter avec une attaque tel que Blackhole. Pour détecter ce type d'attaque qui est considéré comme attaque de topologie RPL, nous avons étudié les traces et le trafic issu de chaque nœud. Afin d'obtenir une analyse complète du comportement du réseau comme nous voulons analyser le comportement des nœuds par rapport à leur position dans l'arbre de routage mais également connaitre combien et où apparaissaient les pertes de paquets. Nous avons donc choisi de mener différentes expériences sur une topologie de plusieurs nœuds sans attaque et une autre étude sur une topologie avec une attaque sinkhole tout en déplaçant le nœuds malicieux dans le réseau.

L'analyse des changements est faites à base des fichiers logs qui trace le trafic sortant d'un nœud. Ces fichiers sont traités par des scripts python pour calculer certain métriques. Nous calculons d'abord taux de paquets reçus (PDR : Packet Delivery Rate) formule 3.5 et le taux de paquets perdus (PLR : Packet Loss Rate) formule 3.6 avant et après l'activation de l'attaque.

Figure 3.1 – *Devenir un routeur important*

$$PDR = \frac{\text{Paquets reçus}}{\text{Paquets envoyés}} \tag{3.5}$$

$$PLR = \frac{\text{Paquets perdus}}{\text{Paquets envoyés}} \tag{3.6}$$

Un autre script nous permet de calculer le nombres de messages de contrôles envoyés pour chaque nœud. Ces messages DIS, DIO, DAO sont à la base de la construction d'un graphe DODAG. Tout changement voir augmentation de ces message indique une instabilité du réseau et des routes. Dans l'analyse des résultats nous avons constaté que le nombre de message DIS ne change pas entre une simulation sans attaque et une autre avec attaque de la même topologie. Par contre, le nombre des message DIO et DAO augmente dans une simulation avec attaque par rapport au même topologie sans attaque. Pour déterminer si cette augmentation est significative, nous avons appliquer la méthode des graphiques de contrôle ou le calcul de seuil 3 sigma. Le seuil des DIO ou DAO est calculer à partir de données issues de simulation où l'attaque est désactivée (sans attaque). Un troisième script pour le calcule du seuil et tracer le graphique des DIO ou DAO dans le cas d'une attaque sinkhole pour voir l'impact de l'attaque sur les nœuds voisin.

3.4. Choix du simulateur

3.4.1. CONTIKI COOJA [14]

Contiki OS est un système d'exploitation open source léger conçu pour l'Internet des Objets. Il a été développé à l'Institut suédois des sciences de l'informatique par Adam Dunkels et écrit dans la langue de programmation C. Contiki est un système d'exploitation hautement portable et il a déjà été porté sur plusieurs plates-formes fonctionnant sur différents types de processeurs. La plupart des plates-formes utilisent le processeur Texas Instruments MSP-430 ainsi que la série de microcontrôleurs Atmel ATmega.

Le principal avantage de Contiki est qu'il fonctionne sur un concept qui se situe entre le multi-threading et la programmation événementielle, cela permet aux processus de partager le même contexte d'exécution et donc d'améliorer l'utilisation de la mémoire et de l'énergie. C'est le concept des Protothreads.

Contiki prend en charge les implémentations de pile IPv6 et IPv4, ainsi que les normes sans fil peu avancées comme 6lowpan, RPL, CoAP ou encore la pile Rime. Il s'agit d'une pile de communication légère pour les réseaux de capteurs et possède des couches plus petites que les piles traditionnelles.

Ce sont des couches simples qui ont de petits en-têtes (seulement quelques octets). Rime prend également en charge la réutilisation du code et le but principal de ce protocole est de simplifier la mise en œuvre des réseaux de capteurs.

Contiki est utilisé dans de nombreux systèmes tels que les compteurs électriques, la surveillance de systèmes industriels, les feux de signalisation, les systèmes d'alarme et la domotique, la surveillance des rayonnements électromagnétique etc. La dernière version de Contiki OS est Contiki 3.0 (publié le 26.08.2015), cela fait donc quelque temps qu'il n'a pas évolué comparé aux possibilités du secteur.

3.4.2. COOJA[14]

Cooja est un outil de Contiki, c'est un simulateur de réseaux de capteurs. C'est une application Java avec une interface graphique (GUI basée sur la trousse à outils Swing standard de Java). Cooja prend en charge la simulation du support radio et l'intégration avec les outils externes pour fournir des fonctionnalités supplémentaires à l'application. Il peut simuler de grands et petits réseaux de différents capteurs qu'on appelle nœuds

sur lesquels on peut charger un système d'exploitation et des applications. Ces capteurs peuvent être émulés à des niveaux plus ou moins complexes.

Cooja comporte deux logiciels émulateurs : Avrora et MSPSim. Cooja utilise Avrora, pour l'émulation des périphériques Atmel AVRbased et MSPSim pour l'émulation de périphériques TI MSP430. La plupart des plates-formes ont des microcontrôleurs MSP430. C'est la raison pour laquelle MSPSim est le logiciel le plus utilisé pour la simulation de réseaux de capteurs sans fil. Cooja peut imiter plusieurs plates-formes comme : TelosB / SkyMote, Zolertia Z1 mote, Wismote, ESB, MicaZ mote. C'est un outil très utile pour le développement et le débogage d'applications Contiki OS. Il permet aux développeurs de tester leur code et leurs systèmes avant de l'exécuter sur le matériel cible réel, d'estimer les consommations d'énergie des nœuds dans les simulations ou de voir les transmissions radio et les réceptions.

3.4.3. Lancer Cooja

Si vous voulez faire une simulation Cooja de RPL, dans l'attaque sinkhole alors les étapes sont les suivantes :

Allez au terminal et tapez : figure 3.2

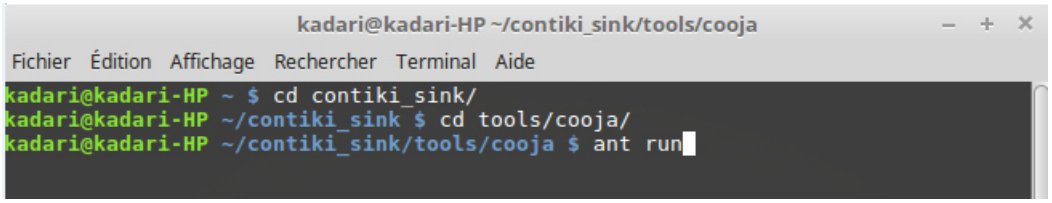

Figure 3.2 – *Lancer Cooja*

pour ouvrir une interface graphique Cooja.

3.4.3.1. Démarrer une simulation

Menu déroulant : File > New simulation (ou raccourci clavier « Ctrl + N »).

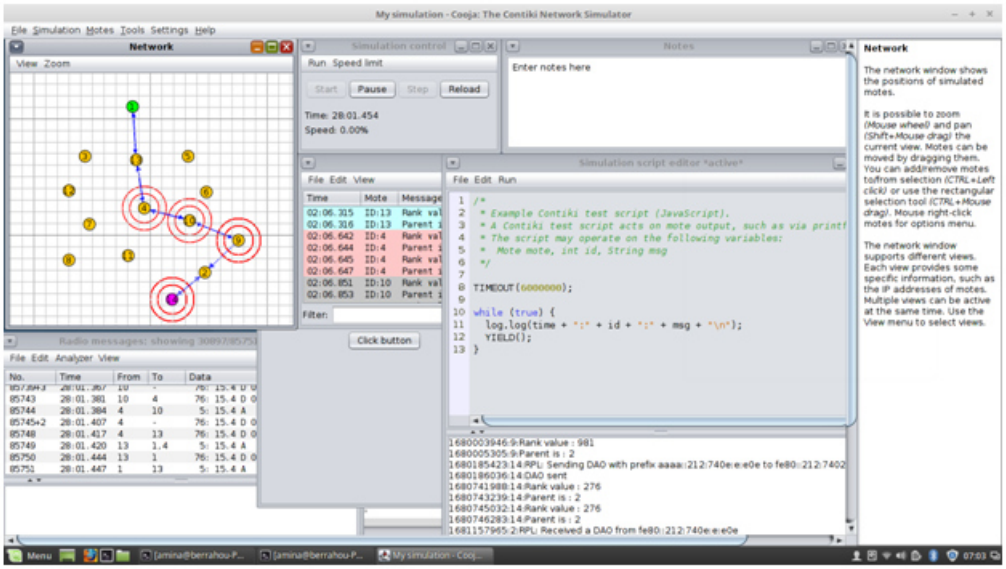

FIGURE 3.3 – *une interface graphique Cooja*

3.4.3.2. AJOUTER UN NŒUD RACINE

On ajoute un « mote » de type « skymote » puis on sélectionne (rpl-collect-sink) le code source udp-sink-c. Ensuite il faut cliquer sur Compiler puis un bouton Créer apparaîtra si la compilation réussit. On peut ensuite ajouter le nombre de nœuds souhaité. Dans notre cas une seule racine.

3.4.3.3. AJOUTER DES NŒUDS « SENDER »

On ajoute un « mote » de type « skymote » puis on sélectionne (rpl-collect-sink) le code source udp-sender.c. On compile le code et on crée plusieurs de ce type, plus ou moins selon la topologie.

3.4.3.4. AJOUTER DES NŒUDS « SINKHOLE »

On ajoute un « mote » de type « skymote » puis on sélectionne (rpl-collect-sink) le code source udp-sender-sinkhole.c. On compile le code et on crée un seul nœud attaquant.

3.5. Déroulement des Simulations

Pour avoir des résultats plus au moins crédibles, et vu le manque d'une base de test fiable pour valider notre approche, nous avons opté à faire une série de simulations pour collecter l'information à analyser. Nous avons réalisé un nombre important de simulation, mais dans notre travail nous présentant les plus importantes pour expliquer notre point de vu.

La série de simulations est principalement regroupée en trois groupe basé sur le nombre de motes (nœuds), pour voir l'impact de l'attaque sinkhole sur l'ensemble des nœuds :

- Groupe 1 : 14 nœuds (un nœud sink, 12 nœuds sender et un nœud sinkhole) ;
- Groupe 2 : 20 nœuds (un nœud sink, 18 nœuds sender et un nœud sinkhole) ;
- Groupe 3 : 50 nœuds (un nœud sink, 48 nœuds sender et un nœud sinkhole).

Dans le premier groupe (14 nœuds) nous avons diversifié les simulations pour étudier l'impact de l'attaque sur la topologie choisi, de ce fait, nous avons traité trois topologies différentes :

- Topologie 1 : le nœud attaquant se positionne au milieu (comme fils de nœuds sender, et père pour d'autres nœuds sender) ;
- Topologie 2 : le nœud attaquant se positionne en bas (étant une feuille de l'arbre, sans avoir de fils nœuds sender) ;
- Topologie 3 : le nœud attaquant se positionne en haut (juste au dessous du nœud sink).

Ce qui nous donne cinq scénarii, et pour chaque scénario on lance la simulation sans activer l'attaque puis on la relance en activant l'attaque sinkhole. Dans chaque cas, on calcule le taux de paquets reçus, le taux de paquets perdus, le nombre de messages de contrôles (DIS, DIO, DAO) envoyés par chaque nœud. La simulation dure aussi longtemps que possible, et après chaque 10 minutes (à savoir, 10mn, 20mn, 30mn, ...) on enregistre le fichier des traces pour faire les calculs sus-mentionnés.

3.5.1. Scénario 1 : 14 nœuds Topologie 1

Dans ce scénario 14 nœuds sont présent selon la topologie 1, Un nœud sink, 12 nœuds sender et un seul nœud sinkhole voir figure 3.4.

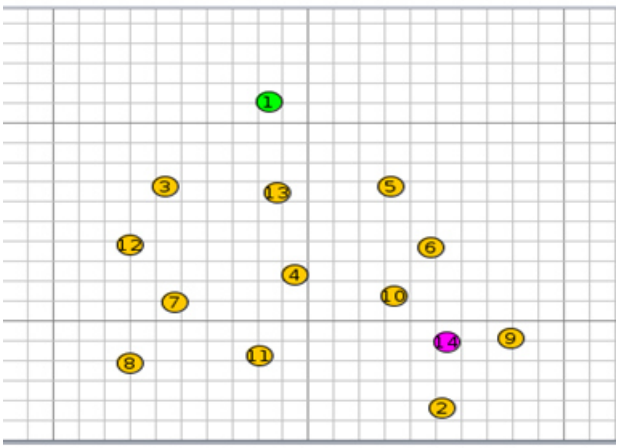

Figure 3.4 – *14 nœuds suivant la topologie 01*

Le tableau suivant 3.1 présente le résultat des simulations après 10 minutes. on remarque qu'il y a une perte de paquets d'environ 30% et une nette augmentation des messages de contrôle DIO et DAO, alors que le nombre de messages DIS ne change pas même après l'activation de l'attaque sinkhole.

Dans cette topologie l'attaque sinkhole ne présente aucun n'impact sur les messages DIS, alors que le nombre de messages DIO et DAO augmente.

10 minute	sans attaque	avec attaque
Total Packet Send	117	117
Total Packet Recv	117	80
PDR	100%	68.38%
PLR	0%	31.62%
DIS	7	7
DIO	155	586
DAO	106	762

Table 3.1 – *14 nœuds, topologie 1 après 10 minutes*

Le tableau 3.2 présente le nombre de messages DIO/DAO délivré par chaque nœud dans le cas où il n'y a pas d'attaque et dans le cas où l'attaque est activée après 10 minutes.

Les Nœuds	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DIO sans attaque	7	11	12	13	11	11	10	11	11	12	12	12	11	11
DIO avec attaque	7	93	13	42	24	61	11	11	94	94	15	11	13	97
DAO sans attaque	0	9	6	7	7	10	10	7	8	10	10	8	5	9
DAO avec attaque	0	138	8	14	10	76	8	7	134	141	9	9	12	196

Table 3.2 – *DIO et DAO pour chaque nœuds après 10 minutes du scénario 1*

Les données issues des lignes sans attaques sont utilisées pour calculer le seuils DIO/DAO par la formule 3.1, et les données issues des lignes avec attaques sont utilisées pour tracer les courbes des DIO voir figure 3.5 et DAO voir figure 3.6, ainsi que la ligne des seuils.

Seuil(DIO) = 13.6

Seuil(DAO) = 12.98

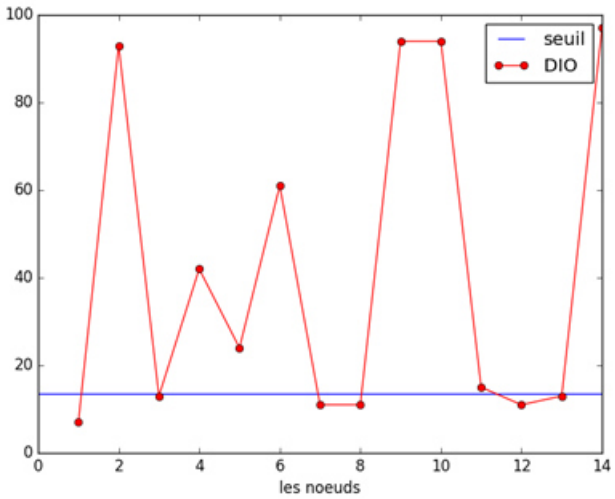

Figure 3.5 – *La Courbe DIO avec attaque après 10 minutes du scénario 1*

On remarque dans les deux courbes que les nœuds 2, 4, 5, 6, 9, 10, 11 et 14 sont au dessus de la ligne du seuil. Sachant que dans cette topologie voir figure 3.4 le nœud malicieux et bien le nœud 14, et les nœuds 2, 4, 5, 6, 9, 10 et 11 qui sont affectés par cette attaque sont au voisinage du nœud attaquant.

On remarque aussi que les nœuds 5 et 11 qui ne sont pas au voisinage très proche de l'attaquant, envoient un nombre élevé de messages DIO que respectivement 24 et 15 au dessus du seuil DIO (13.6), alors leur réponse au messages DIO de leurs voisins par

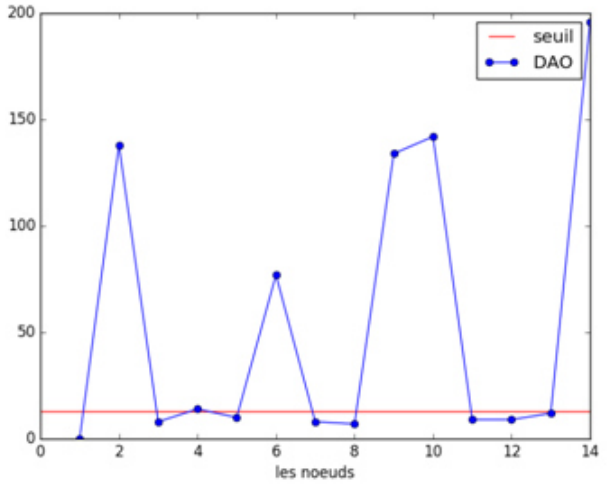

Figure 3.6 – *La Courbe DAO avec attaque après 10 minutes du scénario 1*

des messages DAO à savoir 10 et 9 est au dessous du seuil DAO (12.98).

Dans le même scénario on a laissé la simulation se dérouler jusqu'à environ 20 minutes pour voir l'impact de l'attaque dans le temps. les résultats sont présent dans le tableau 3.3.

20 minute	sans attaque	avec attaque
Total Packet Send	242	240
Total Packet Recv	242	171
PDR	100%	71.25%
PLR	0%	28.75%
DIS	7	7
DIO	235	1134
DAO	132	1414

Table 3.3 – *14 nœuds, topologie 1 après 20 minutes*

Le tableau 3.4 présente le nombre de messages DIO/DAO délivré par chaque nœud dans le cas où il n'y a pas d'attaque et dans le cas où l'attaque est activée après 20 minutes.

Seuil(DIO) = 19.99

Seuil(DAO) = 16.58

Les Nœuds	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DIO sans attaque	8	18	17	18	18	16	17	18	18	19	17	18	16	17
DIO avec attaque	8	186	18	95	37	119	17	17	189	189	25	18	17	199
DAO sans attaque	0	11	8	8	8	12	12	9	10	12	13	10	6	13
DAO avec attaque	0	264	9	22	11	104	12	11	260	287	27	11	19	377

Table 3.4 – *DIO et DAO pour chaque nœuds après 20 minutes du scénario 1*

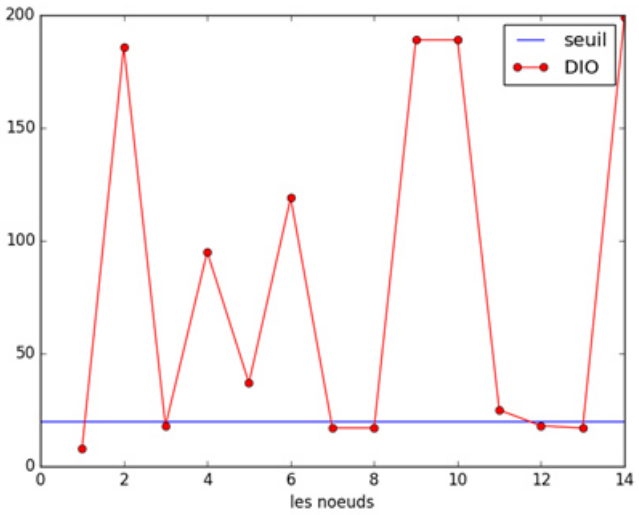

Figure 3.7 – *La Courbe DIO avec attaque après 20 minutes du scénario 1*

Les courbes des DIO voir figure 3.7 et DAO voir figure 3.8, ainsi que la ligne des seuils, montrent qu'après 20 minutes les même nœuds 2, 4, 5, 6, 9, 10, 11 et 14 restent au dessus du seuil DIO. Le nœud 5 répond par des messages DAO (11) moins que le seuil DAO (16.58), alors que le nœud 11 (27 message DAO) passe au dessus du seuil, de même le nœud 13 (19 DAO).

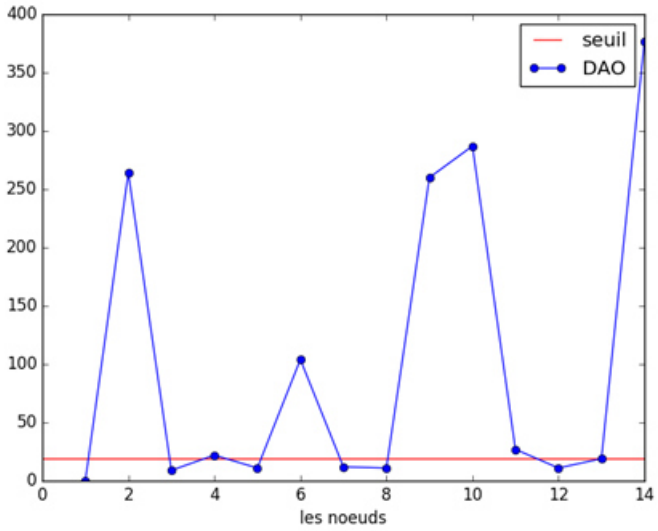

FIGURE 3.8 – *La Courbe DAO avec attaque après 20 minutes du scénario 1*

3.5.2. SCÉNARIO 2 : 14 NŒUDS TOPOLOGIE 2

Dans ce scénario 14 nœuds sont présent selon la topologie 2, Un nœud sink, 12 nœuds sender et un seul nœud sinkhole voir figure 3.9, le nœud malicieux est en bas de l'arbre.

Le tableau 3.5 présente le résultat des simulations après 10 minutes. on remarque qu'il y a une perte de paquets d'environ 13% et une nette augmentation des messages de contrôle DIO et DAO, alors que le nombre de messages DIS ne change pas même après l'activation de l'attaque sinkhole.

Dans la topologie 2, comme dans la topologie 1, l'attaque sinkhole ne présente aucun n'impact sur les messages DIS, alors que le nombre de messages DIO et DAO augmente.

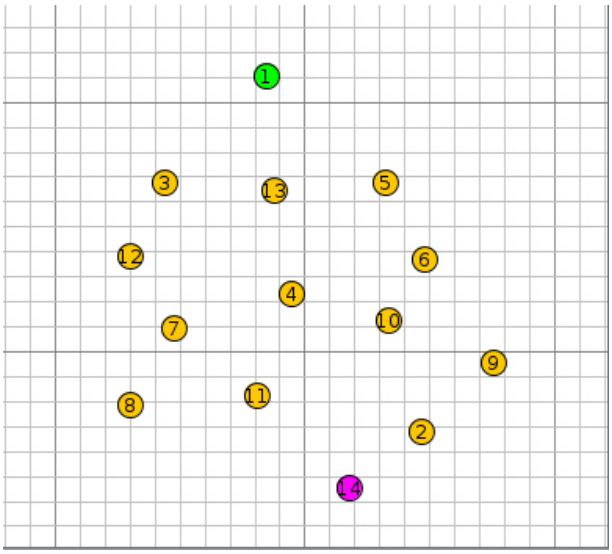

FIGURE 3.9 – *14 nœuds suivant la topologie 2*

10 minute	sans attaque	avec attaque
Total Packet Send	117	117
Total Packet Recv	117	102
PDR	100%	87.18%
PLR	0%	12.82%
DIS	7	7
DIO	165	429
DAO	113	412

TABLE 3.5 – *14 nœuds, topologie 2 après 10 minutes*

Le tableau 3.6 présente le nombre de messages DIO/DAO délivré par chaque nœud dans le cas où il n'y a pas d'attaque et dans le cas où l'attaque est activée après 10 minutes.

De même, que dans le scénario 1, dans ce scénario les données issues des lignes sans attaques sont utilisées pour calculer le seuils DIO/DAO par la formule 3.1, et les données issues des lignes avec attaques sont utilisées pour tracer les courbes des DIO voir figure 3.10 et DAO voir figure 3.11, ainsi que la ligne des seuils.

Seuil(DIO) = 17.40

Seuil(DAO) = 13.46

On remarque dans la courbe des DIO figure 3.10, que les nœuds 2, 9, 10 et 14 sont au dessus de la ligne du seuil. Sachant que dans cette topologie voir figure 3.9 le nœud

Les Nœuds	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DIO sans attaque	7	13	11	12	12	10	17	11	11	14	12	13	11	11
DIO avec attaque	7	93	12	12	12	11	11	11	78	64	11	11	11	85
DAO sans attaque	0	9	8	7	10	9	10	12	7	10	6	9	7	9
DAO avec attaque	0	159	7	6	9	8	9	8	85	11	5	9	6	90

Table 3.6 – *DIO et DAO pour chaque nœuds après 10 minutes du scénario 2*

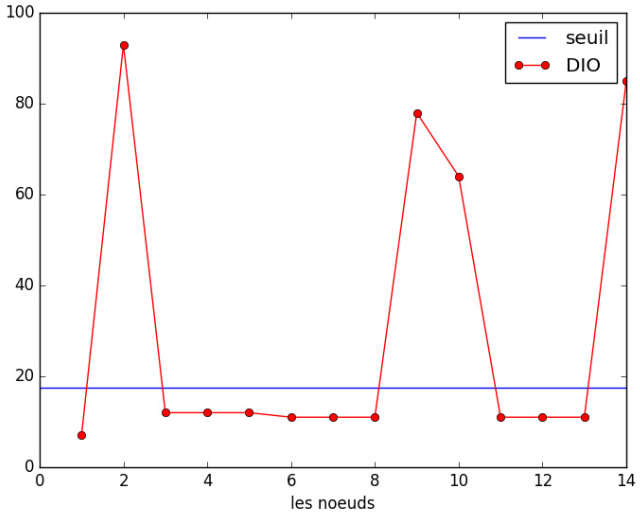

Figure 3.10 – *La Courbe DIO avec attaque après 10 minutes du scénario 2*

malicieux et bien le nœud 14, et les nœuds 2, 9 et 10 qui sont affectés par cette attaque sont en lien du nœud attaquant.

Dans la courbe des DAO figure 3.11, que les nœuds 2 et 9 qui sont plus proche du nœud attaquant répondent par des messages DAO plus que le seuil (13.46).

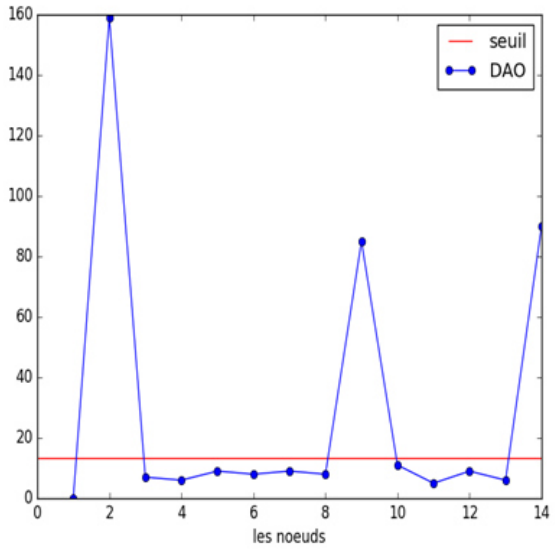

FIGURE 3.11 – *La Courbe DAO avec attaque après 10 minutes du scénario 2*

Dans le même scénario 2 on a laissé la simulation se dérouler jusqu'à environ 20 minutes pour voir l'impact de l'attaque dans le temps dans une autre topologie. les résultats sont présent dans le tableau 3.7.

20 minute	sans attaque	avec attaque
Total Packet Send	247	247
Total Packet Recv	247	215
PDR	100%	87.04%
PLR	0%	12.96%
DIS	7	7
DIO	239	828
DAO	144	848

TABLE 3.7 – *14 nœuds, topologie 2 après 20 minutes*

Le tableau 3.8 présente le nombre de messages DIO/DAO délivré par chaque nœud dans le cas où il n'y a pas d'attaque et dans le cas où l'attaque est activée après 20 minutes.

Seuil(DIO) = 22.04

Seuil(DAO) = 17.68

Les Nœuds	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DIO sans attaque	8	18	17	18	17	17	22	17	17	19	18	18	17	16
DIO avec attaque	8	192	18	24	18	24	16	18	156	122	17	17	18	180
DAO sans attaque	0	11	10	9	10	10	14	16	10	13	9	13	8	11
DAO avec attaque	0	341	9	8	10	9	13	10	197	28	12	12	7	192

Table 3.8 – *DIO et DAO pour chaque nœuds après 20 minutes du scénario 2*

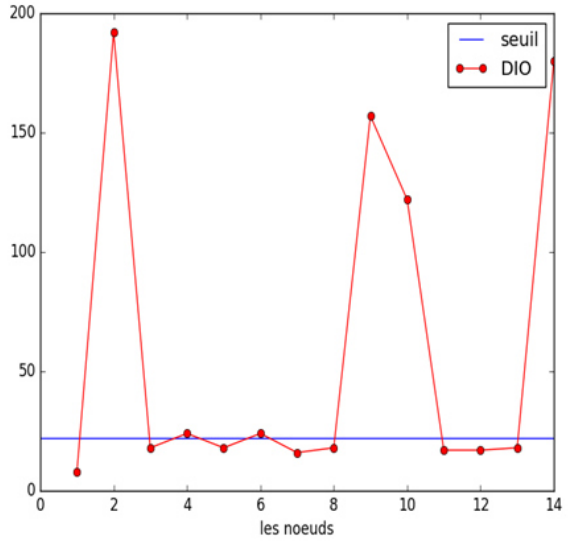

Figure 3.12 – *La Courbe DIO avec attaque après 20 minutes du scénario 2*

On remarque dans la courbe des DIO figure 3.12, que les mêmes nœuds 2, 9, 10 et 14 sont au dessus de la ligne du seuil, après 20 minutes deux autres nœuds ont dépassé le Seuil (22.04), le nœud 4 et le nœud 6 avec 24 messages DIO pour chacun. On remarque que l'effet de l'attaque s'est propagé vers ces deux nœuds dans cette topologie voir figure 3.9.

Dans la courbe des DAO figure 3.13, que les nœuds 2, 9 et le nœud 10 qui s'ajoute à la liste des nœuds qui ont répondu par des messages DAO plus que le seuil (17.68).

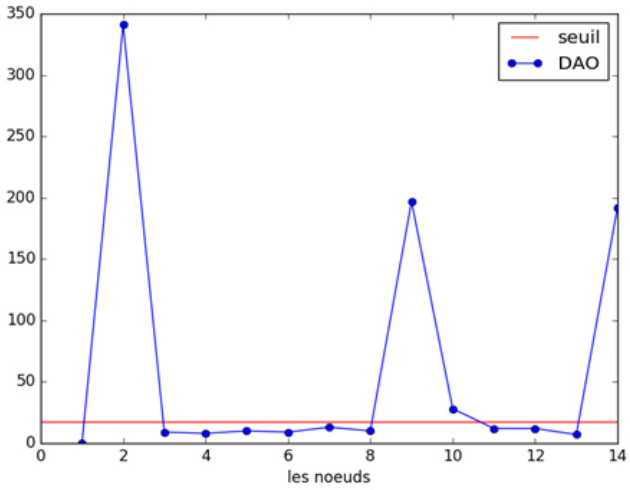

FIGURE 3.13 – *La Courbe DAO avec attaque après 20 minutes du scénario 2*

3.5.3. SCÉNARIO 3 : 14 NŒUDS TOPOLOGIE 3

Dans ce scénario 14 nœuds sont présent selon la topologie 3, Un nœud sink, 12 nœuds sender et un seul nœud sinkhole voir figure 3.14.

Dans ce scénario on a laissé la simulation aller jusqu'à 1 heure, vu que même dans le cas de l'activation de l'attaque sinkhole aucun paquet n'est perdu et le nombre des messages de contrôle DIS, DIO et DAO n'a pas changé, comme il paraît dans le tableau 3.9.

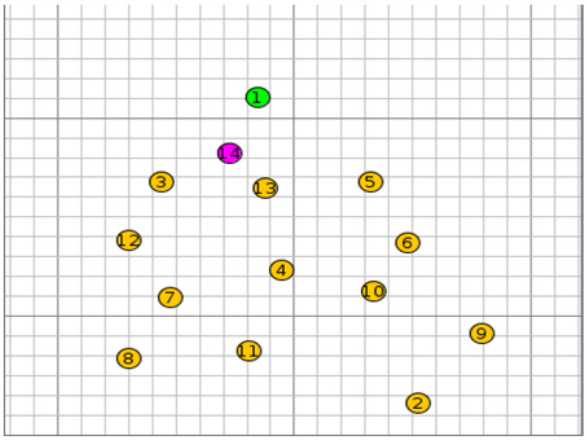

FIGURE 3.14 – *Topologie du scénario 3*

1 heure	sans attaque	avec attaque
Total Packet Send	767	767
Total Packet Recv	767	767
PDR	100%	100%
PLR	0%	0%
DIS	6	6
DIO	532	532
DAO	201	201

TABLE 3.9 – *14 nœuds, topologie 3 après 1 heure*

Seuil(DIO) = 46.36
Seuil(DAO) = 25.96

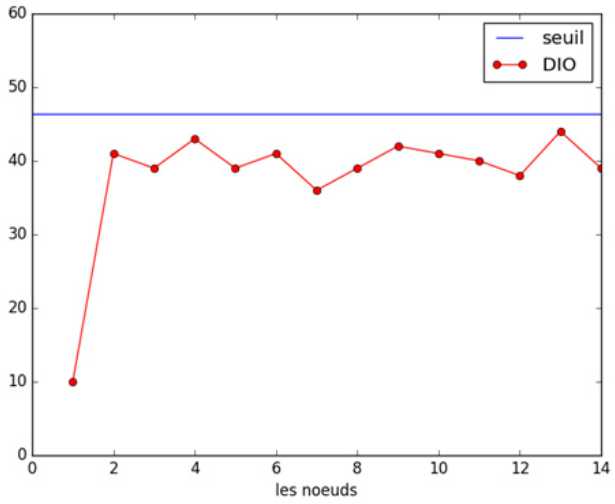

FIGURE 3.15 – *La Courbe DIO avec attaque aprés 1heure*

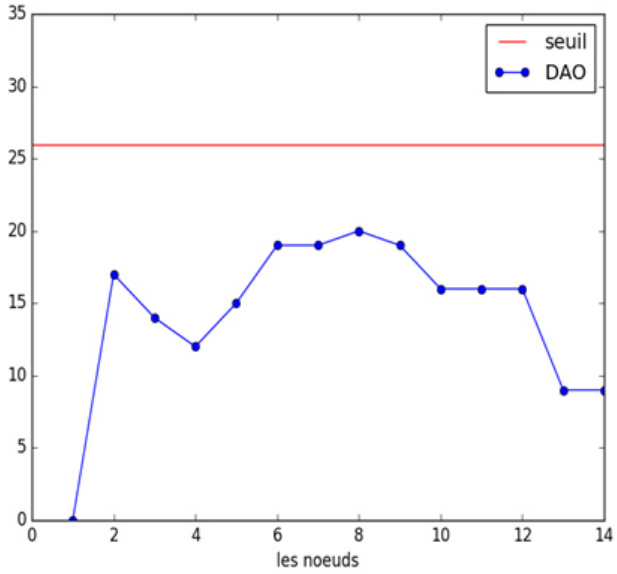

Figure 3.16 – *La Courbe DAO avec attaque aprés 1heure*

D'après les calcules effectués, le nœud malicieux ne présente aucun danger dans cette topologie il n'y a eu aucun changement de DIO, DAO. Les courbes des DIO figure 3.15 et des DAO figure 3.16 montre bien que tous les nœuds sont au dessous du seuil DIO/DAO.

3.5.4. Scénario 4 : 20 nœuds

Dans ce scénario 20 nœuds sont présent selon une topologie quelconque, Un nœud sink, 18 nœuds sender et un seul nœud sinkhole voir figure 3.17.

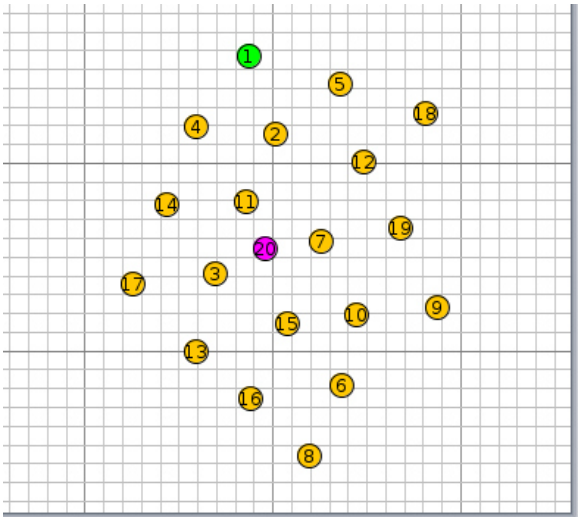

Figure 3.17 – *Topologie du scénario 4*

Le tableau suivant 3.10 présente le résultat des simulations après 10 minutes. on remarque qu'il y a une perte de paquets d'environ 5% et une nette augmentation des messages de contrôle DIO et DAO, alors que le nombre de messages DIS ne change pas même après l'activation de l'attaque sinkhole.

10 minute	sans attaque	avec attaque
Total Packet Send	171	171
Total Packet Recv	171	163
PDR	100%	95.32%
PLR	0%	4.68%
DIS	11	11
DIO	230	397
DAO	157	345

Table 3.10 – *Scénario 4 : 20 nœuds après 10 minutes*

Le tableau 3.11 présente le nombre de messages DIO/DAO délivré par chaque nœud dans le cas où il n'y a pas d'attaque et dans le cas où l'attaque est activée après 10 minutes pour les 20 nœuds.

Les Nœuds	1	2	3	4	5	6	7	8	9	10
DIO sans attaque	7	12	11	12	11	11	12	12	12	13
DIO avec attaque	7	18	31	19	12	21	23	27	18	21
DAO sans attaque	0	7	9	9	9	5	6	7	10	9
DAO avec attaque	0	6	26	6	10	18	27	25	18	22
Les Nœuds	11	12	13	14	15	16	17	18	19	20
DIO sans attaque	11	11	11	11	11	12	12	12	14	12
DIO avec attaque	23	13	23	16	25	21	15	12	20	32
DAO sans attaque	7	8	7	8	10	12	7	9	11	7
DAO avec attaque	30	6	19	10	34	24	11	8	13	32

Table 3.11 – *DIO et DAO pour chaque nœuds après 10 minutes du scénario 4*

Les données issues des lignes sans attaques sont utilisées pour calculer le seuils DIO/DAO par la formule 3.1, et les données issues des lignes avec attaques sont utilisées pour tracer les courbes des DIO voir figure 3.18 et DAO voir figure 3.19, ainsi que la ligne des seuils.

Seuil(DIO) = 14.09

Seuil(DAO) = 13.4

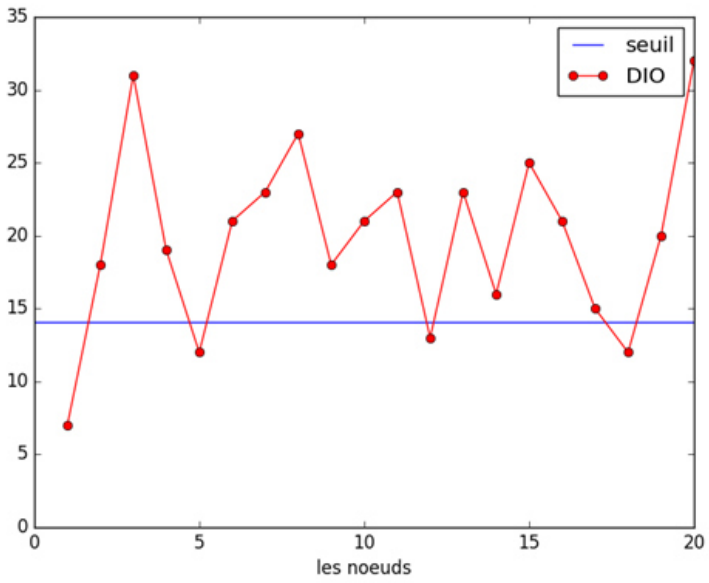

Figure 3.18 – *La Courbe DIO avec attaque aprés 10 minute*

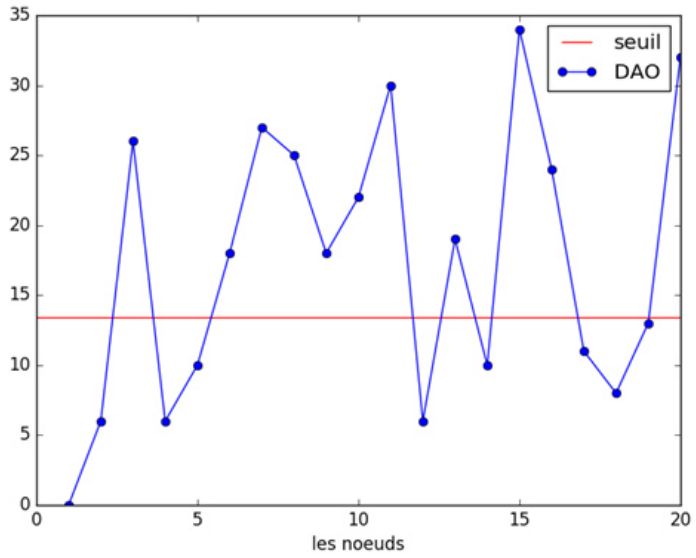

Figure 3.19 – *La Courbe DAO avec attaque aprés 10 minute*

Les courbes des DIO voir figure 3.18 et DAO voir figure 3.19, ainsi que la ligne des seuils, montrent qu'après 10 minutes que tous les nœuds sauf les nœuds 5, 12 et 18 sont au dessus du seuil DIO (14.09). Alors que les nœuds 3, 6, 7, 8, 9, 10, 11, 13, 15, 16 et 20 répondent par des messages DAO plus que le seuil DAO (13.4).

Dans le même scénario 4 on a laissé la simulation se dérouler jusqu'à environ 20 minutes pour voir l'impact de l'attaque dans le temps. les résultats sont présent dans le tableau 3.12.

20 minutes	sans attaque	avec attaque
Total Packet Send	321	320
Total Packet Recv	321	308
PDR	100%	96.25%
PLR	0%	3.75%
DIS	11	11
DIO	328	601
DAO	186	508

Table 3.12 – *Scénario 4 : 20 nœuds après 20 minutes*

Le tableau 3.13 présente le nombre de messages DIO/DAO délivré par chaque nœud dans le cas où il n'y a pas d'attaque et dans le cas où l'attaque est activée après 20

minutes.

Seuil(DIO) = 21

Seuil(DAO) = 15.6

Les Nœuds	1	2	3	4	5	6	7	8	9	10
DIO sans attaque	8	17	17	17	17	17	16	16	16	18
DIO avec attaque	7	27	45	27	17	35	37	37	31	34
DAO sans attaque	0	8	10	9	10	7	8	8	14	10
DAO avec attaque	0	7	35	9	14	32	43	31	23	34
Les Nœuds	11	12	13	14	15	16	17	18	19	20
DIO sans attaque	17	15	16	16	16	17	17	16	22	17
DIO avec attaque	38	17	31	33	39	29	25	17	28	47
DAO sans attaque	9	10	8	10	12	14	9	10	12	8
DAO avec attaque	39	10	26	25	49	34	18	10	16	53

TABLE 3.13 – *DIO et DAO pour chaque nœuds après 20 minutes du scénario 4*

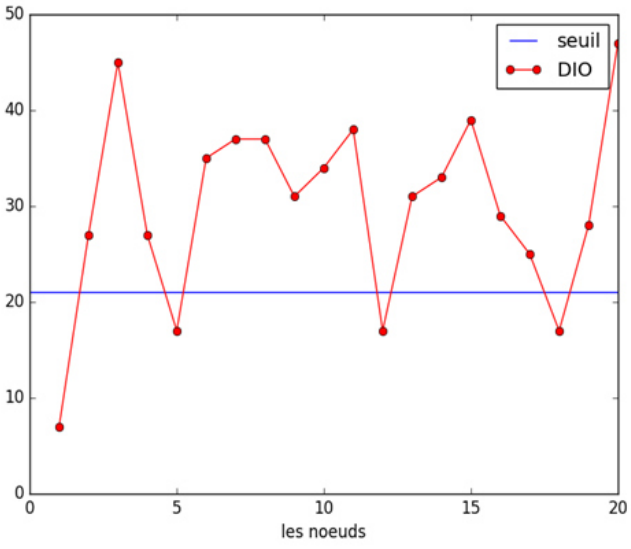

FIGURE 3.20 – *La Courbe DIO avec attaque après 20 minutes*

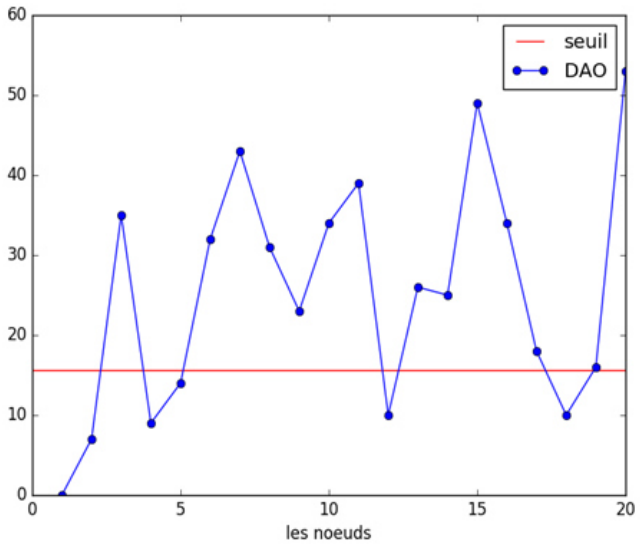

Figure 3.21 – *La Courbe DAO avec attaque après 20 minutes*

Après 20 minutes, on peut remarquer d'après les courbes des DIO voir figure 3.20 et DAO voir figure 3.21, que la ligne des seuils, montrent que les même qui était au dessus du seuil DIO (14.09) de 10 minutes, restent toujours au dessus du seuil DIO (21) de 20 minutes. Les même nœuds aussi 3, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16 et 20 répondent par des messages DAO plus que le seuil DAO (15.6). Ce qui explique que dans le temps les mêmes nœuds sont affectés.

3.5.5. Scénario 5 : 50 nœuds

Une topologie d'un nœud sink, 48 Noueds Sender, 1 Noeud Sender-sinkhole, suivant cette topologie voir figure 3.22. On réalise une simulation sans activer l'attaque mais ne dépasse pas 2 :04 (2minute), puis activé l'attaque et la simulation ne dépasse pas 1mn :30. Les résultats sont présenté ci-dessous :

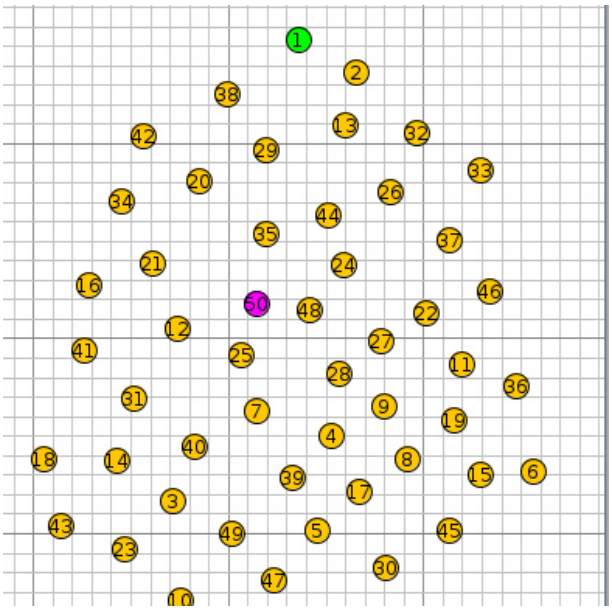

FIGURE 3.22 – *Topologie de 50 Noeuds*

Sans attaque, 2 :04 minute :

Total Packet Send = 50

Total Packet Recv = 37

Time = 124

PDR = 74.0 %

PLR = 26.0 %

Total Packet DAO = 319

Total Packet DIS = 49

Total Packet DIO = 366

Avec attaque, 1 :25 minute :
Total Packet Send = 9
Total Packet Recv = 5
Time = 76
PDR = 55.56 %
PLR = 44.44 %
Total Packet DAO = 244
Total Packet DIS = 49
Total Packet DIO = 263

La courte durée de cette simulation nous n'a pas permis de voir l'effet de l'attaque sinkhole, quoique le nombre élevé des messages DIO/DAO dans le cas de l'attaque pendant 1mn :25sec, comparé au nombre des message dans le cas sans attaque qui a durée 2mn :04sec, et le taux de perte qui passe de 26% à 44%, peut signifié le dommage que cause cette attaque dans tel cas.

3.6. Conclusion

Dans ce chapitre nous avons effectué une série de simulations pour voir l'impact de l'attaque sinkhole dans un réseau RPL. Nous nous sommes basé sur l'étude statistique (méthode de Shewhart) des messages de contrôle (DIO/DAO) à partir des traces des nœuds. Nous avons réalisé un nombre important de simulations pour étudier l'impact de cette attaque, mais dans ce mémoire, nous avons choisi uniquement cinq scénarios qui nous paraissent représentatifs.

Dans les scénarios 1 et 2, l'attaque sinkhole représente un danger pour ce type de réseau, du moment que le nombre de messages de contrôle DIO/DAO dépasse de manière importante le normale. Cette augmentation a un effet sur tout le réseau qui sera pénalisé par l'épuisement des ressources des nœuds qui sont au voisinage du nœud malicieux. Dans le cas des simulations sans attaque, la durée pourrait allé jusqu'à des heures, alors que si on active l'attaque, la simulation ne dure pas plus que 20 minutes.

Le scénario 3, nous a permis de constater que dans les topologies où le nœud attaquant est un fils directe du nœud sink (root), cette attaque n'a aucun effet sur le réseau RPL, ni en perte de paquets ni en nombre de messages de contrôle envoyé, et

par conséquence même dans le cas d'activation de l'attaque la simulation peut durée longtemps.

Dans les autres scénarios, l'impact de l'attaque sinkhole dans un réseau comporte un nombre important de nœuds.

Conclusion générale

Dans ce travail, nous avons étudié l'impact de l'attaque sinkhole dans un réseau RPL. Dans l'Internet des Objets le protocole RPL est un des principales protocoles de routage. La nature des objets et les contraintes qui posent, sont décisif pour le choix de la méthode de détection des attaques.

Notre approche pour étudié l'attaque sinkhole dans un réseau RPL se base sur les graphiques de contrôle (Control Chart) de Walter A. Shewhart. Nous avons pu déterminer, en réalisant des simulation, l'impact de cette attaque. Le calcul statistique nous a permis de distinguer les nœuds affectés par le nœud malicieux. Nous avons exploité les fichiers de traces pour calculer le nombre de message de contrôle (DIO/DAO).

L'attaque sinkhole a un effet remarquable sur un réseau RPL dans le temps est quelques soit le nombre de nœuds. Dans plusieurs topologies, où le nœud malicieux est au milieu ou en bas de l'arbre, l'attaque a des effets sur le réseau, mais dans le cas où l'attaquant est un fils directe du sink, l'attaque ne pourra pas être détectée ce qui nous donne des faux positif.

Nous espérons approfondir cette étude et ajouter d'autre critère pour détecter l'existence de l'attaque sinkhole et aller jusqu'à la détection du noeud malicieux en étalant l'approche pour étudier l'historique des rangs et la liste des parents d'un nœud.

Table des figures

Liste des tableaux

Bibliographie

[1] Somia SAHRAOUI Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things) 23 Nov 2016 thèse de doctorat -batna.

[2] Les Réseaux de capteurs sans Fils (WSN : Wireless Sensor Networks) caracteristique des reseau wsn -Yacine CHALLAL Université de Technologie de Compiègne, FRANCE

[3] Simulation of a Monitoring scheme in Internet of Things UNIVERSITE D'AVIGNON Mr. BENSLIMANE Abderrahim mai 2017 pp9-13

[4] Résilience et application aux protocoles de routage dans les réseaux de capteurs pp3-4 Lyon 2013 pp14-20

[5] sécurisation d'un protocole inter-couche pour les reseaux LR-WPAN these de doctorat louazani ahmed, mai 2015 universite d'oran pp24-27

[6] Gestion de risques appliquée aux réseaux RPL Anthéa Mayzaud,Isabelle Chrisment, pp2-4

[7] Les Réseaux de capteurs sans Fils (WSN : Wireless Sensor Networks)- caracteristique des reseau wsn -Yacine CHALLAL Université de Technologie de Compiègne, FRANCE https ://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_24.html

[8] Anthéa Mayzaud Monitoring and Security for the RPL-based Internet of Things Supervisions sécurité pour l'Internet des Objets utilisant le protocole de routage RPL 21/10/2016(taxonomie des attaques)pp9-14

[9] Gestion de risques appliquée aux réseaux RPL - Anthéa Mayzaud- pp3-4 Université de Lorraine https ://sarssi14.liris.cnrs.fr/ressources/pdfs/sarssi2014_amayzaud.pdf

[10] L. Atzori, A. Lera, G. Morabioto, The Internet of Things : a survey, Computer Networks 54 (15) (2010) 2787–2805.

[11] Abdelmalek Boudries, Maintien de la Connectivité dans les Réseaux Ad hoc sans fil 2014, PP. 4-9

[12] ETUDE DU RSSI POUR L'ESTIMATION DE LA DISTANCE DANS LES RESEAUX DE CAPTEURS SANS FIL SARI Mounya Amal Master en Informatique juin 2017 Protocoles de support IPv6 pour réseaux de capteurs sur courant porteur en ligne Cédric Chauvenet 2015 lien : https ://tel.archives-ouvertes.fr/tel-01168472/document

[13] Protocoles de support IPv6 pour réseaux de capteurs sur courant porteur en ligne Cédric Chauvenet 2015 lien : https ://tel.archives-ouvertes.fr/tel-01168472/document.

[14] Simulation of a Monitoring scheme in Internet of Things pp17-18 BENSLIMANE Abderrahim mai2017 Projet Master