République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa

Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Master Recherche
Option : Réseau et Sécurité RN

Thème

Etude d'impacts des attaques sur le protocole de routage RPL dans l'IoT

Présenté par :

KHALDI Tinhinane KHELIFI Cylia

Soutenu devant le jury composé de :

Président	Dr AKILAL Abdellah	M.C.A	U. A/Mira Béjaïa
Examintaeur	Dr FARAH Zoubeyr	M.C.A	U. A/Mira Béjaïa
Encadrant	Dr MOKTEFI Mohand	M.C.A	U. A/Mira Béjaïa

Promotion 2022 – 2023

Dédicace

Je dédie cet événement marquant dans ma vie à la mémoire de mon cher grand père, que dieu l'accueil dans son vaste paradis,

À mes très chers parents, maman et papa pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études, que dieu leur donne une longue et joyeuse vie ainsi à Imalili et ma grand-mère,

À mon cher et unique frère Smail, à mes chères sœurs Farida, Souad et Nabila pour leurs encouragements permanents,

À mon neveu Yanis, et mes petites nièces Sidra, Miral, Alae, Mayline,Eline et Amélia,

À mes deux oncles, mes tantes, mes proches et tout mes cousins et cousines en particulier Lila, Asma, Feriel,

À tout mes professeurs tout au long de mon parcours scolaire et universitaire,

À mes amis et collègues qui m'ont toujours encouragés, et à qui je souhaite plus de succès surtout mon amie tinhinane,

Au final, je le dédie pour ma chère amie, ma binôme Cylia, Merci d'avoir été une collègue formidable, je souhaite que l'amitié qui nous a réunit persiste pour toujours et que nous arriverons à réaliser nos rêves.

Merci pour tout !

- Tinhinane

Dédicace

“

Je dédie ce travail à ma mère, qui a été mon pilier et ma source d'inspiration tout au long de ma vie. Ton amour inconditionnel et ton soutien constant ont été mes plus grands atouts. Merci d'avoir été là à chaque étape de mon parcours, en m'encourageant et en me poussant à donner le meilleur de moi-même,

À mon père, dont l'intégrité et la détermination ont été des exemples inspirants. Ton soutien infaillible, tes encouragements et tes précieux conseils m'ont guidé sur le chemin de la réussite, Je vous aime de tout mon cœur et je dédie ce mémoire à vous, mes merveilleux parents,

À Mes chères sœurs Milissa et Kamelia, qui ont toujours été là pour moi. Je vous dédie ce mémoire en reconnaissance de notre amour fraternel qui ne cesse de grandir,

À mes amis les plus chers et mes cousins, qui ont illuminé mon parcours académique de leur présence,

À mon cher mari, mon soutien inébranlable et mon partenaire de vie tout au long de cette aventure. Tu as été là m'encourageant à poursuivre mes rêves, me donnant la force de continuer même dans les moments les plus difficiles. Merci d'avoir cru en moi, de m'avoir soutenu sans relâche et d'avoir sacrifié tant de temps et d'efforts pour que je puisse me consacrer à mes études. Ce mémoire

est dédié à toi, pour tout ce que tu représentes dans ma vie et pour l'amour infini que nous partageons,

À ma belle-famille, pour votre soutien précieux et votre amour inconditionnel tout au long de ce parcours,

ma binôme Tinhinane, mon complice dans cette aventure académique. Ensemble, nous avons relevé les défis, surmonté les obstacles. Notre partenariat a été une source d'inspiration et de motivation, et je suis reconnaissant(e) d'avoir eu la chance de travailler à tes côtés.

Merci pour tout !

”

- Cylia

Remerciements

Tout d'abord, nous remercions ALLAH le tout-puissant de nous avoir donné la santé, le courage et la patience nécessaires à mener ce travail à son terme.

Nous remercierons et témoignons notre reconnaissance à notre directeur de mémoire **M.Mohand MOKTEFI**, pour son soutien, ses précieux conseils et son expertise tout au long de ce projet. Ses orientations éclairées et sa disponibilité ont grandement contribué à l'aboutissement de ce travail de recherche.

Nos remerciements vont également à tous les membres de jury, pour avoir accepté de consacrer leur temps et leur expertise à l'évaluation de ce mémoire.

On tient à exprimer notre reconnaissance envers nos professeurs qui nous ont transmis leurs connaissances et leur passion pour le domaine des réseaux. Leurs enseignements ont été d'une grande valeur pour notre formation académique et ont guidé nos réflexions tout au long de ce mémoire.

Nos vifs remerciements à nos familles pour leur amour, leur soutien indéfectible et leurs encouragements constants. Leur soutien inconditionnel a été une source d'inspiration et de motivation tout au long de ce mémoire.

On souhaite également adresser nos remerciements à nos collègues et amis qui nous ont soutenu et encouragé tout au long de ce parcours académique.

Pour finir, nous souhaitons remercier chaleureusement toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire. Leur soutien, leurs conseils et leur présence ont été inestimables et ont joué un rôle déterminant dans la réussite de ce travail de recherche.

Merci infiniment à tous.

Liste des sigles et acronymes

6LoWPAN	*ipv6 Low power Wireless Personal Area Network*
DAO	*Destination advertisement Object*
DAO-ACK	*Destination advertisement Object Acknowledgement*
DIO	*DODAG information Object*
DIS	*DODAG information solicitation*
DODAG	*Destination Oriented Directed Acyclic Graph*
ETX	*Expected Transmission Count*
IEEE	*Institute of Electrical and Electronics Engineers*
IETF	*Internet Engineering Task Force*
IoT	*Internet of things*
IPv6	*Internet protocol Version 6*
LLN	*Low power and lossy network*
UDP	*User datagram protocol*
OF	*Objective Function*
OF0	*Objective Function Zero*
PDR	*Packet Delivery Rate*
RFC	*Request For Comments*
ROLL	*Routing Over Low-power and Lossy*

RPL *Routing Protocol for Low-Power*

WSN *Wireless Sensor Neeworks*

Introduction générale

Contexte

L'internet et le réseau informatique ont connu un énorme développement ce qui a fait naître l'internet des objets (connu sous le nom de IoT : Internet of things) et grâce à son avènement, de nombreux dispositifs connectés sont déployés pour faciliter notre vie quotidienne. Cependant, la gestion des communications et du routage au sein de ces réseaux d'objets pose des défis uniques. Le protocole de routage RPL (Routing Protocol for Low-Power and Lossy Networks) a été spécialement conçu pour répondre aux exigences des réseaux à faible consommation d'énergie et présentant des pertes de paquets.

Le protocole de routage RPL est largement utilisé dans les réseaux IoT pour permettre une connectivité efficace et fiable entre les nœuds. Il utilise une structure hiérarchique appelée graphe DODAG (Destination Oriented Directed Acyclic Graph) pour organiser les nœuds en fonction de leurs relations de parenté. Cela permet d'optimiser les performances du réseau en termes de consommation d'énergie, de latence et de fiabilité des communications.

Problématique

Avec la popularité croissante des réseaux IoT, les attaques ciblant le protocole de routage RPL sont devenues une préoccupation majeure. Les attaquants peuvent exploiter les vulnérabilités du protocole pour perturber le routage, compromettre la sécurité des données et épuiser les ressources des nœuds. Comprendre l'impact de ces attaques sur les réseaux IoT est essentiel pour développer des mécanismes de défense éfficaces et garantir la fiabilité et la sécurité des communications.

Dans ce mémoire, nous nous concentrerons sur l'analyse de l'impact des attaques sur le protocole de routage RPL dans les réseaux IoT. Nous examinerons les différentes attaques possibles, telles que les attaques de ressources et les attaques d'altération de la topologie. Nous évaluerons l'effet de ces attaques sur les performances du réseau, notamment la consommation d'énergie, la surcharge du trafic et la fiabilité des communications.

Objectifs

L'objectif principal de ce mémoire est de fournir des connaissances approfondies sur les vulnérabilités du protocole de routage RPL dans l'IoT et d'implémenter diverses attaques afin d'évaluer le protocole RPL. Nous aborderons des sujets tels que les vulnérabilités du RPL et l'évaluation des impacts.

Organisation du mémoire

Pour mener à bien notre recherche, le présent travail est organisé en quatre chapitres selon le plan méthodologique suivant, la partie théorique se compose de trois chapitres :

Le premier chapitre, intitulé "**Généralités IoT** " fait l'objet de la présentation des réseaux IoT, son architecture et sa pile protocolaire.

Le deuxième chapitre, intitulé "**Étude des attaques RPL**" est une présentation détaillée du protocole de routage RPL qui fait l'objet de notre étude, et ses différentes attaques ainsi que les contre-mesures existantes pour faire face à ces menaces.

Le troisième chapitre, intitulé "**État de l'art**" qui est une revue de littérature sur les attaques IoT, précisemment les attaques RPL.

La partie pratique, comprenant un seul chapitre : au titre de " **Simulation** " sera consacré à la simulation des quatres attaques de routage. Nous allons présenter du pseudo-code pour chaque attaque, et une évaluation des performances de RPL sera présentée selon deux métriques de routage. Cette évaluation est suivie d'une discussion des résultats obtenus.

Et enfin nous terminerons avec une conclusion générale, ainsi que quelques perspectives pour des travaux futurs.

Chapitre 1

Généralités IoT

1.1 Introduction

L'Internet des Objets (IoT) a émergé comme une révolution technologique majeure, ouvrant de nouvelles perspectives pour la connectivité et l'interaction entre les objets physiques et le monde numérique. L'IoT se réfère à un réseau interconnecté d'appareils, de capteurs et d'actuateurs qui peuvent collecter, échanger et agir sur des données via des connexions Internet. Ce chapitre vise à fournir une vue d'ensemble des généralités sur l'IoT. Tout d'abord, nous aborderons les principes fondamentaux de l'IoT tels que la définition de cette nouvelle technologi, qui a pu marquer son utilisation dans divers domaines, son historique, puis nous allons voir son architecture, fonctionnements et les différents protocoles de l'IoT et enfin, nous allons finir par une conclusion.

1.2 L'internet des Objets (IdO)

Plusieurs définitions ont été données à l'internet des objets ou "Internet of Things (IoT)" en anglais,Étant donné la complexité de l'Internet des objets (IdO), il est difficile de le définir de manière exhaustive en une seule définition. Certaines définitions mettent l'accent sur les aspects techniques de l'IdO, tandis que d'autres se concentrent davantage sur les utilisations et les fonctionnalités. Dans ce qui suit nous allons présenter ces différentes définitions données à l'IoT : ce terme a été défini par différents auteurs de différentes manières. L'Internet des objets (IoT) est défini comme un paradigme dans lequel des objets équipés de capteurs, d'actionneurs et de processeurs communiquent entre eux pour servir un objectif utile [49].

L'IoT peut se définir aussi comme étant "un réseau de réseaux qui permet, via des systèmes d'identification électroniques normalisés et unifies, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi, de pouvoir récupérer, stocker, transférer et traiter les données sans discontinuité entre les mondes physiques et virtuels" [16].

L'IoT donc fait référence à un réseau d'appareils physiques, de véhicules, de bâtiments et d'autres objets qui sont intégrés avec des capteurs, des logiciels et une connectivité réseau, leur permettant de connecter et d'échanger des données sur internet en suivant les protocoles qui assurent leur communication et échange d'informations à travers une variété de dispositifs.

L'Internet des objets (IdO) simplifie notre vie en automatisant des tâches et en permettant aux objets connectés de collecter et d'échanger des données. Les applications de l'IdO sont vastes, allant des machines à café automatiques à la production industrielle, en passant par les dispositifs de santé intelligents. En combinant l'IdO avec l'intelligence artificielle, les objets connectés fonctionnent de manière fiable et rapide.

1.3 L'objet connecté

Un objet connecté (OC) est un dispositif dont la principale fonction n'est pas d'être un système informatique ou une interface d'accès à Internet. Par exemple, une machine à café ou une serrure traditionnelle ont été conçues sans avoir intégrer de systèmes informatiques ou de connexion à Internet. Cependant, en ajoutant une connexion Internet à un OC, il devient un OC enrichi (OCE), ce qui lui permet d'offrir davantage de fonctionnalités et d'interagir avec son environnement. Un OC peut interagir avec le monde physique de manière autonome, sans intervention humaine. Il est soumis à diverses contraintes telles que la mémoire, la bande passante ou la consommation d'énergie, qui doivent être prises en compte lors de sa conception [13].

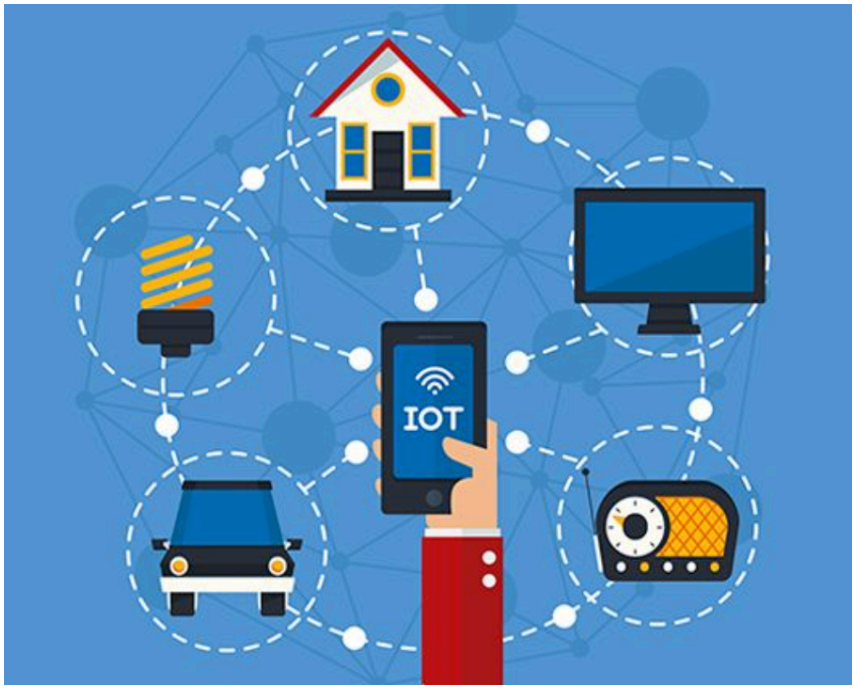

Fig. 1.1 : Objet connecté [1]

1.4 Historique

Historiquement, c'est en 1998 que Kevin Ashton, gestionnaire de marque chez Procter et Gamble(directeur exécutif de MIT Auto-ID Labs), a employé le terme "Internet des objets" pour la première fois pour décrire la connexion entre la technologie RFID et l'Internet.Il mentionna que : l'Internet des objets a le potentiel de changer le monde, tout comme Internet l'a fait. Peut-être encore plus" ensuite, il déclara lors d'une réunion du groupe : "si nous parvenons à ajouter l'identification par fréquence radio et d'autres capteurs aux objets de la vie quotidienne, nous pourrons alors créer un Internet des Objets et poser les fondations d'une nouvelle ère de la perception par les machines" [27].

Ce concept s'est développé de manière impressionnante en intégrant constamment de nouvelles technologies et des objets innovants. Cela implique la création d'un réseau

mondial d'objets interconnectés, adressables de manière unique, en utilisant des protocoles de communication standard.

1.5 Les composants de l'IoT

Les composants de l'iot sont cinq, la composante principale de l'IoT est l'objet connecté, qui peut être conçu pour être connectable ou avoir une connectivité ajoutée ultérieurement. L'objet collecte et traite des données de capteurs, les communique et reçoit des instructions pour exécuter une action.Pour ces fonctions, une source d'énergie est généralement nécessaire, surtout si les données sont prétraitées dans l'objet [12]

1. **Les capteurs :**Un capteur IoT est un dispositif électronique qui transforme une information physique (température, pression, débit...) en un signal électronique, il permet de collecter ces données physiques et les envoyer ensuite à un serveur ou une plateforme IoT pour être analysées, traitées et utilisées pour prendre des décisions ou pour automatiser des systèmes.

   Exemples de capteurs : position, proximité, déplacement, accélération, lumière, température, humidité, son, vibration, chimique, gaz, flux, pression, etc.

2. **Les actionneurs :** L'actionneur est un dispositif matériel qui permet de transformer une information digitale en un phénomène physique, d'où sa dénomination. Un actionneur IoT permet de contrôler ou de modifier l'état d'un objet physique dans le monde réel, en fonction des données reçues à partir d'autres dispositifs IoT.

   Exemple d'actionneurs : Afficheurs, Alarmes, Caméras, Haut-parleurs, Interrupteurs, Lampes, Moteurs, Pompes, Serrures, Vannes, Ventilateur, etc.

3. **Énergie :** La contrainte la plus importante à laquelle les capteurs sont soumis est l'énergie. l'autonomie des nœuds est évaluée en termes d'années.

4. **Réseau de capteurs :** Les capteurs sont équipés de dispositifs sans fil pour émettre et recevoir des données, mais cela ne suffit pas pour rendre un ensemble de capteurs accessible et interopérable. Pour cela, les capteurs doivent s'organiser en un réseau de capteurs, qui est caractérisé par des éléments très petits avec des capacités de transmission sans fil.

5. **La connectivité :** Les objets IoT ont une antenne RF pour se connecter aux réseaux et transmettre des informations telles que leur identité, leur état, des alertes et des données de capteurs. De plus, ils peuvent recevoir les données et des commandes en retour. Le module de connectivité est essentiel à la gestion du cycle de vie de l'objet.

1.6 Domaines d'application

L'IdO s'est déployé dans de nombreuses applications, comme le montre la figure 3. Elles sont devenues intelligentes et effectuent leur travail de manière robotisée en s'appuyant

sur l'internet qui améliorerait le genre de nos vies personnelles, des entreprises et des communautés. Qui touche essentiellement : la domotique, les villes, le transport, la santé et l'industrie. Dans ce qui suit, nous présentons les domaines d'application les plus pertinents.

1.6.1 La domotique et les villes intelligentes :

La domotique : ou maison connectée, représente l'utilisation de l'Internet des Objets dans une maison. Grâce à la domotique, Les humains utilisent de nombreux appareils électroniques comme : les réfrigérateurs, les fours à micro-ondes, les ventilateurs, les chauffages et les climatiseurs à la maison. Les capteurs sont installés pour détecter les problèmes et les communiquer à l'entreprise de fabrication afin qu'elle les résolve.

Les smart cities : Il existe aussi les villes intelligentes qui permettent d'améliorer la qualité de vie des habitants, en assurant une consommation de ressources minimales grâce à une combinaison intelligente des infrastructures (énergie, transport, communication) aux différents niveaux hiérarchiques (ville, quartier, bâtiment). La ville de Santander, en Espagne, a été l'une des premières, en Europe, à se lancer dans une stratégie smart city à grande échelle.

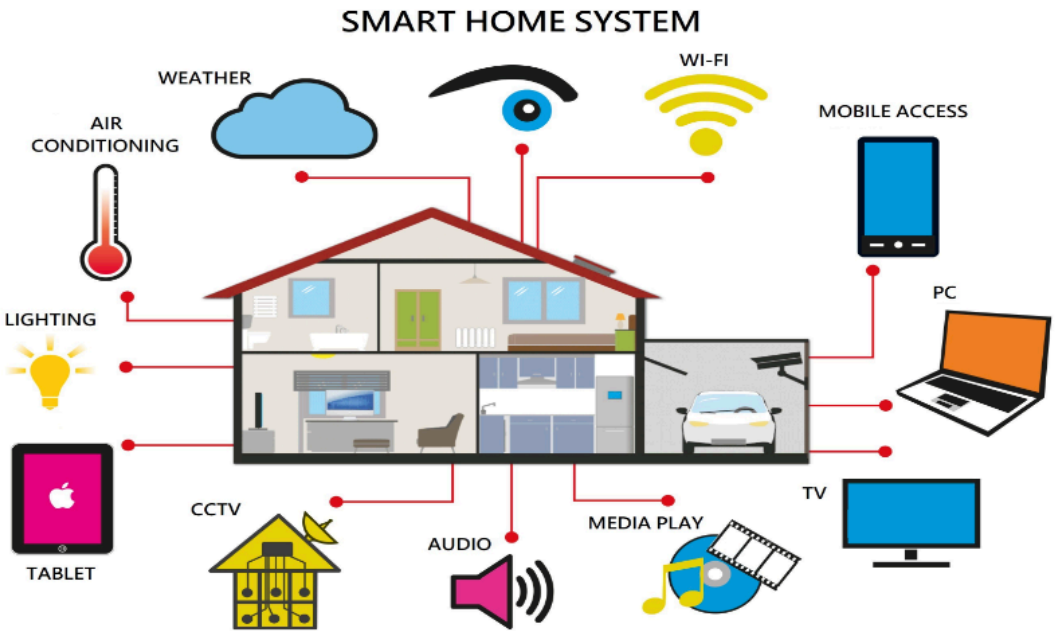

FIG. 1.2 : Système de maison intelligente [2]

1.6.2 Le transport :

Il existe plusieurs moyens de transport tels que les voitures, les trains, les bus, etc. Sont actuellement devenus intelligents en leur équipant par des capteurs, actionneurs et une puissance de traitement. L'IoT peut sauver des vies, car il peut réduire le trafic et minimiser l'impact des véhicules sur l'environnement.

Les voitures d'aujourd'hui évoluent pour devenir de véritables ordinateurs qui progressent vers la conduite autonome, comme les véhicules actuellement testés par Google. Bien que nos voitures ne soient pas encore totalement autonomes, elles deviennent de plus en plus autonomes grâce à des systèmes d'automatisation de certaines tâches de conduite, telles que l'allumage des phares ou le freinage automatique.

1.6.3 La santé (Smart Health)

Dans le domaine de la santé, l'IoT joue un rôle essentiel, il permettra le déploiement de réseaux personnels pour le contrôle et le suivi des signes cliniques, les objets connectés permettent de suivre la tension, le rythme cardiaque, la qualité de respiration ou encore la masse graisseuse grâce à des capteurs et des puces. Ceci permettra ainsi de faciliter la télésurveillance des patients à domiciles notamment pour des personnes âgées pour les éviter de se déplacer jusqu'aux hôpitaux en particulier lorsqu'il s'agit d'une maladie très contagieuse telle que le COVID-19, la télémédecine est très recommandée. De nos jours, il existe aussi ce qu'on appelle les hôpitaux intelligents qui sont dotés de nouvelles technologies.

1.6.4 L'industrie

Grâce à la technologie IoT, il sera possible d'assurer un suivi complet des produits, de la chaîne de production jusqu'à la chaîne logistique et de distribution, en surveillant les conditions d'approvisionnement. Cela permet aux usines d'améliorer l'efficacité de ses opérations, d'optimiser la production et d'améliorer la sécurité des employés.

1.6.5 L'énergie

L'IoT facilite l'échange d'informations en temps réel entre les nombreux appareils du réseau électrique, permettant ainsi une distribution et une gestion de l'énergie plus efficace.

1.6.6 Le militaire

Les objets connectés reliés à Internet offrent de nouvelles opportunités pour une utilisation militaire. En exploitant des technologies innovantes, ils pourraient fournir des solutions utiles dans le domaine militaire. Dans le domaine militaire, une application pertinente consiste à déployer un réseau de capteurs dans des zones stratégiques ou difficiles d'accès pour surveiller les activités des forces ennemies. Cependant, il est crucial de garantir une cybersécurité solide pour assurer l'efficacité de ces systèmes. Des essais concluants ont déjà été menés dans ce domaine par l'armée américaine [38].

1.6.7 L'agriculture

Il utilise les technologies IoT qui permettent une gestion efficace des ressources. Cela peut aider à surveiller le développement des plantes et contrôler en cas de changements drastiques inattendu de l'évolution due à la température ou l'humidité. À l'aide de drones munis d'un système GPS, les exploitants peuvent ainsi recueillir en temps réel des données ou des images pour vérifier l'humidité du sol ou encore l'état des cultures et des plantations.

1.6.8 L'éducation

Dans les années à venir, la majorité des établissements d'enseignement adopteront progressivement les technologies de l'IoT afin d'améliorer les processus d'apprentissage et l'enseignement à distance. En appliquant l'IoT dans l'éducation, l'efficacité opérationnelle de l'école, la sécurité du campus et la qualité de l'éducation peut être améliorée [3].

Voici quelques exemples concrets des applications de l'Internet des objets (IdO) dans le domaine de l'éducation :

- Surveillance des présences.
- Tableaux blancs intelligents et autres médias numériques interactifs.
- Sécurité sur le campus.
- Des cartes d'étudiant intelligentes.
- Capteurs de température et de l'équipement pour le chauffage, la ventilation et l'air conditionné pour réduire la consommation d'énergie [23].

1.6.9 Le sport

L'utilisation de l'iot dans le domaine sportif est de plus en plus courante, car elle a permis de développer des objets connectés pour les sportifs. Le premier objet qui vient à l'esprit est désormais le bracelet connecté pour but de donner les statistiques à des sessions de sport. D'une façon générale, on peut imaginer que n'importe quel accessoire de sport pourrait devenir un objet connecte comme par exemple : ballons connectés en Bluetooth qui calculent la puissance de frappe, la trajectoire et la rotation, des raquettes de tennis connectées, etc.

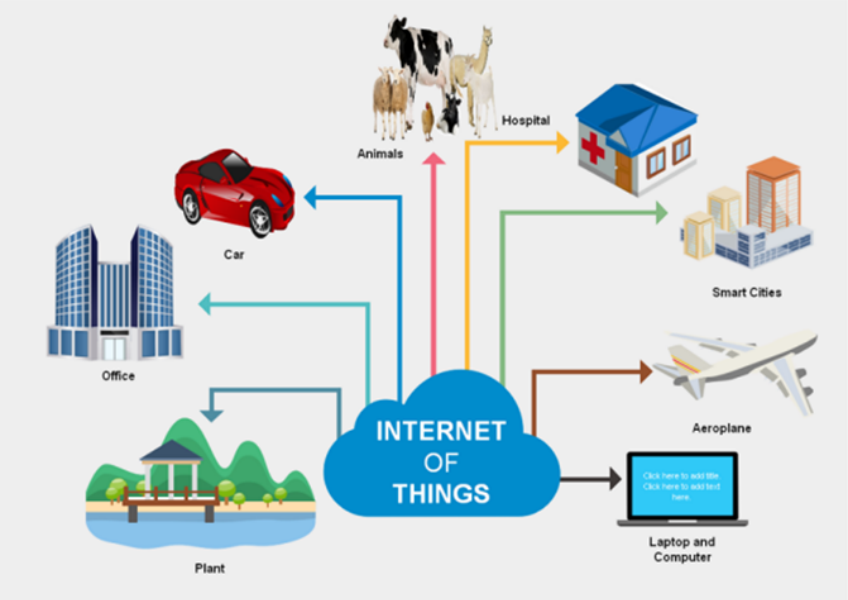

FIG. 1.3 : Domaines d'application [18]

1.7 Technologies de l'Internet des Objets

Un système IoT réunit de nombreux acteurs et composants technologiques qui assurent le bon fonctionnement d'un système IoT.

En effet, bien qu'il existe plusieurs technologies utilisées dans le fonctionnement de l'IoT, nous mettons l'accent seulement sur trois technologies clé, le RFID, WSN et M2M.

1.7.1 RFID

La RFID est une technologie sans fil qui est utilisée pour l'identification des objets, dans laquelle une étiquette RFID (une petite puce avec une antenne) transporte des données qui sont lues par un lecteur RFID. Elle englobe toutes les technologies qui utilisent les ondes radio pour identifier automatiquement des objets ou des personnes. RFID utilise un ensemble de bandes de fréquences (125 kHz, 13,56 MHz, 433 MHz, 865-868 MHz, 2,45-5,8 GHz et 3,1-10 GHz). Sa portée varie selon la fréquence, elle peut aller jusqu'à des centaines de mètres [13].

1.7.2 WSN

Un réseau de capteurs WSN (Wireless Sensor Network) est un réseau composé d'un ensemble de nœuds-capteurs, ces derniers sont capables de collecter, traiter, analyser et disséminer des informations et communiquent via des ondes radio afin de surveiller des phénomènes précis. Un nœud capteur est composé généralement d'interfaces de capture de l'information, d'un microprocesseur, d'une unité mémoire, d'une interface de communication et d'une batterie comme source d'énergie [18].

1.7.3 M2M

C'est l'association des technologies de l'information et de la communication avec des objets intelligents. Permettant une communication entre différentes machines ou équipements (véhicules, capteurs, caméras) connectés à différents réseaux sans l'intervention de l'humain. Les applications M2M utilisent différentes interfaces et protocoles pour les échanges entre les différents réseaux mobiles par exemple.

1.8 Architecture générale de l'IoT

- **La couche de perception :** elle représente la couche physique, car la tâche principale de la couche de perception est de reconnaître les propriétés physiques telles que la température, l'humidité, le niveau de la lumière, la vitesse, etc. À l'aide, des capteurs et des actionneurs afin de recueillir des informations sur l'environnement, ces informations seront envoyées à la couche suivante [17].

- **La couche réseau :** également connue sous le nom de "couche de transmission" est une couche responsable de la connexion et la transmission des données reçues de la couche de perception. Elle est également utilisée pour transmettre et traiter les données des capteurs. Les principales technologies utilisées pour réaliser cette couche sont : les technologies cellulaires, WiFi, Bluetooth, Zigbee [17].

- **La couche d'application :** est la responsable de la gestion des interactions directe avec les utilisateurs finaux. Elle traite les données reçues de la couche réseau et elle est chargée de fournir à l'utilisateur des services spécifiques et applications intelligentes [17].

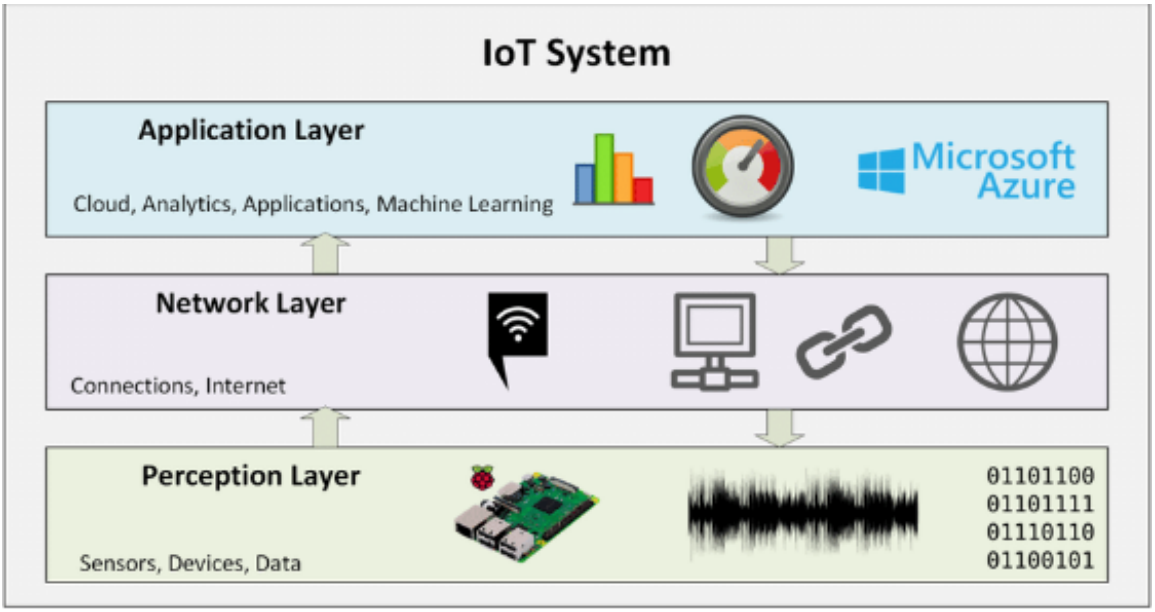

Fig. 1.4 : Architecture générale de l'IoT [8].

- **Architecture à quatre couches**

  En raison du développement continu de l'Internet des objets, toutes les exigences de l'Internet des objets ne peuvent pas être satisfaites, pour cela une architecture a quatre couches a été proposé, elle comporte trois couches comme l'architecture précédente, mais elle comporte également une couche supplémentaire appelée couche de support. Le but de créer cette couche est la sécurité.
  Dans une architecture à quatre couches, les informations sont envoyées à la couche network obtenue à partir de la couche perception. La couche de support a deux responsabilités. Elle confirme que les informations sont envoyées par un utilisateur réel et utilise des méthodes d'authentification pour prévenir les menaces. La deuxième responsabilité est d'envoyer des informations à la couche réseau [17].

- **Architecture à cinq couches**

  L'architecture à quatre niveaux a joué un rôle important dans le l'évolution de l'IoT. L'architecture à quatre niveaux présentait également des problèmes de sécurité et de stockage. Les chercheurs ont proposé une architecture à cinq niveaux sont, la couche de perception, couche de transport, couche d'application de plus couche processing et couche Business [17].

  - **La chouche processing** : elle est connue sous le nom de couche middleware et joue le rôle de collecter les informations provenant de la couche de transport. Elle traite ensuite ces informations collectées et a la responsabilité de supprimer les données superflues [17].
  - **La couche business** : sa responsabilité principale est de prendre en charge la gestion et le contrôle des applications. De plus, il possède la capacité de déterminer les modalités de création, de stockage et de modification des informations [17].

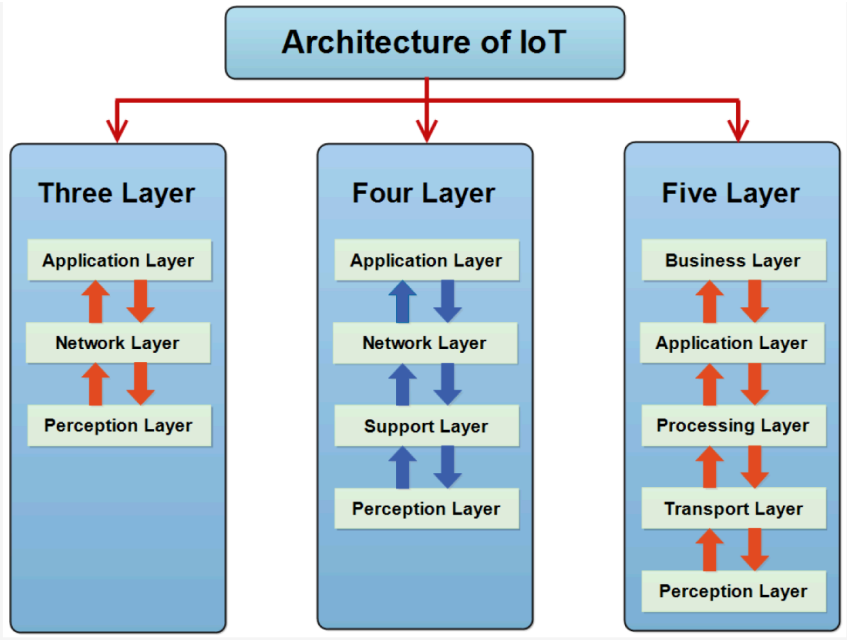

FIG. 1.5 : Les architectures en couches de l'IoT (trois, quatre et cinq couches) [17].

1.9 Connectivité

Pour que les objets connectés communiquent entre eux, les protocoles de communication qui sont différents seront utilisés selon le réseau auquel l'objet appartient (longue et courte portée), ces protocoles désignent des modes de communication permettant d'assurer une sécurité optimale des données échangées entre les appareils connectés à l'iot. Ils autorisent la connexion des appareils IoT via un réseau IP ou non IP. Bien qu'il existe une différence dans leurs portées, leur consommation d'énergie et de mémoire. Les principaux protocoles utilisés comprennent :

1.9.1 Protocoles de réseaux IoT

La norme IEEE 802.15.1 / Bluetooth : Inventé en 1994 par la société suédoise Ericsson, le protocole Bluetooth est un standard de transfert de données sans fil. Elle a été standardisée sous la norme IEEE 802.15.1, il utilise une faible bande passante, ce qui ne lui permet de transférer que peu de données à de courtes distances. Inclus à l'immense majorité des téléphones mobiles, afin de réaliser une communication entre deux téléphones, ou entre un téléphone et un objet connecté de nature différente, il possède désormais de nombreuses applications : oreillette de discussion téléphonique sans fil, montre intelligente, moniteur de fréquence cardiaque, station météo, thermostat, domotique intelligente et un système de surveillance du trafic [47].

La norme IEEE 802.15.4/ Zigbee : Egalement connue sous le nom de Zigbee, est spécialement conçue pour les réseaux à dimension personnelle (Wireless Personal Area Networks : WPANs). il permet la transmission de données sur de plus longues distances tout en offrant une faible consommation d'énergie, On retrouve ZigBee dans les contrôles

industriels, les applications médicales, les détecteurs de fumée, etc [47].. Elle offre la possibilité de créer des topologies réseau comprenant un très grand nombre de capteurs.

La norme IEEE 802.11x/WiFi : Le Wi-Fi regroupe un ensemble de protocoles de communication sans fil qui permettent des connexions haut débit sur des distances allant de 20 à 100 mètres. Cependant, il est important de noter que le Wi-Fi est un réseau local sans fil qui consomme beaucoup d'énergie, ce qui le rend plus adapté aux appareils alimentés sur secteur ou ayant une alimentation électrique régulière. Il offre la possibilité de transférer rapidement de grandes quantités de données [47].

Les protocoles de réseaux GSM : Fournis par les opérateurs de télécommunication, les réseaux cellulaires mobiles, basés sur la technologie GSM, permettent de transférer une quantité importante de données à une longue portée. Ils nécessitent l'installation d'une carte SIM dans l'appareil à connecter, afin d'identifier celui-ci sur le réseau de communication. Succédant aux premières générations des standards pour la téléphonie mobile, qui ont progressivement permis d'accroître le débit de communication, la quatrième génération (4G) permet une communication mobile à très haut débit.

SigFox : SigFox est une technologie de communication sans fil à faible consommation d'énergie conçue pour les objets à faible consommation comme les capteurs et les applications M2M. Elle permet de transférer de petites quantités de données sur des distances allant jusqu'à 50 kilomètres. Cette technologie est largement utilisée dans divers domaines tels que les compteurs intelligents, les moniteurs de patients, l'agriculture, les dispositifs de sécurité, l'éclairage public et les capteurs environnementaux [47].

Lora : LoRa (Long Range) est un protocole de communication sans fil conçue pour fournir les réseaux étendus de faible consommation et grande portée et une transmission de données sécurisée. La norme LoRa a été développée pour les dispositifs de type IoT dans les réseaux régionaux ou mondiaux [46].

1.9.2 Protocoles applicatifs

Est un ensemble de règles définissant le mode de communication entre deux applications informatiques. Les protocoles de la couche application sont utilisés pour échanger des données entre les programmes s'exécutant sur les hôtes source et de destination. Il existe des protocoles de la couche application qui permettent la communication dans l'IoT [21].

- MQTT (Message Queuing Telemetry Transport)

- CoAP (Constrained Application Protocol)

- AMQP (Advanced Message Queuing Protocol)
- XMPP (Protocole de messagerie et de présence extensible)
- DDS(Data Distribution Service)

1.10 IPv6 et les IoT

L'Internet des objets (IoT) englobe un nombre extrêmement élevé de nœuds, et il est essentiel que chaque nœud soit identifiable et accessible par tout utilisateur autorisé, peu importe sa position. Pour résoudre ce défi, l'utilisation de l'adressage IPv6 a été proposée pour l'IoT. Les adresses IPv6 sont représentées par des chiffres binaires de 128 bits, offrant ainsi une capacité suffisante pour identifier tous les objets qui nécessitent une adresse.

1.11 La pile protocolaire de l'IoT

Différentes entreprises et organisations ont proposé plusieurs piles protocolaires pour l'IoT. La littérature présente des piles protocolaires constituées de trois couches : la couche de détection, la couche réseaux et communications et la couche d'applications, comme proposé par Minerva et al.[19]. En revanche, Granjal et al.[26] ont proposé une pile protocolaire plus complète constituée de cinq couches : la couche physique, la couche MAC, la couche adaptation, la couche réseau et routage, et la couche d'applications.

Dans le contexte des objets connectés, les interfaces de communication sans fil de type 802.15.4 sont couramment utilisées. Ces protocoles radio ont une consommation énergétique très faible, ce qui permet de constituer des réseaux de faible puissance appelés LLN (Low-Power and Lossy Networks) ou LowPAN (LoW Power wireless Area Networks).

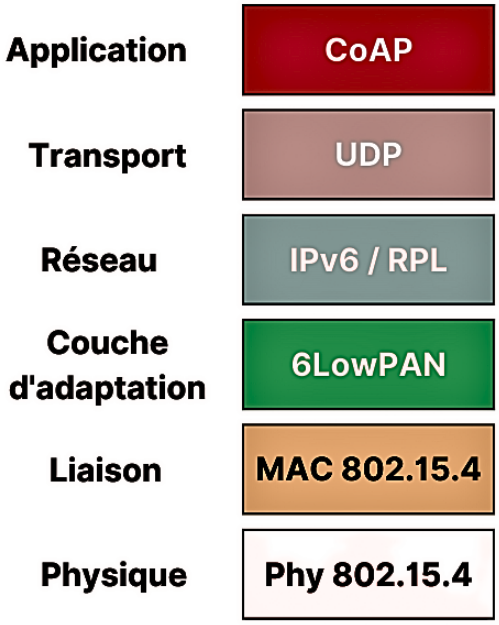

FIG. 1.6 : la pile protocolaire de l'IoT [4]

.

Comme le montre la figure ci-dessus le réseau IoT est constitué d'une pile de six couches. Les couches "physique" et "MAC" sont supportées par les normes IEEE 802.15.4 et IEEE 802.15.4e. La couche adaptation utilise 6LoWPAN et la couche réseau utilise principalement le protocole RPL pour le routage. La couche application utilise CoAP pour le transfert web optimisé.

1.11.1 la couche physique

La couche physique est chargée de gérer le support physique des transmissions. Elle détermine les méthodes de conversion des bits en signaux analogiques et inversement.

1.11.2 La couche MAC

En plus de gérer la consommation d'énergie, considérée comme l'une des fonctions primordiales dans les réseaux de capteurs sans fil, la couche MAC est également responsable des tâches suivantes : les confirmations de réception, les créneaux horaires dédiés, la découverte des voisins, les balises, l'accès au canal physique, la validation des trames, l'association des nœuds et la sécurité, qui n'est disponible que dans cette couche selon la norme IEEE 802.15 [26].

1.11.3 La couche adaptation

La couche d'adaptation [6LowPAN] permet de faire passer des trames IPv6 dont le MTU est au minimum de 1280 octets dans des trames 802.15.4 (127 octets). Elle est conçue pour faciliter l'interopérabilité entre différents types de réseaux en IPv6 tels que l'Ethernet, le 802.15.4, le Wifi, la 4G, etc. Cette technologie permet de supporter les communications Internet pour l'IoT et standardise les protocoles de communication IoT ainsi que les communications de bout en bout entre les dispositifs capteurs IoT.

1.11.4 La couche réseau

La communication entre les nœuds du réseau est gérée par le protocole RPL, qui est un protocole de routage à vecteur de distance adapté aux réseaux peu fiables caractérisés par des pertes de paquets et une faible bande passante. Dans les LLN, où les pertes de paquets sont tolérées, les données sont généralement transportées par UDP, ce qui permet de limiter la taille des en-têtes [49].

1.11.5 La couche application

Le protocole CoAP (Constrained Application Protocol) est utilisé pour les communications de la couche application dans l'IoT, en permettant l'interopérabilité avec l'architecture Web et les communications de bout en bout entre les périphériques IoT et les autres entités Internet. Ce protocole est actuellement limité aux communications UDP sur 6LoWPAN [49].

1.12 Sécurité dans l'IoT

Étant donné les problèmes de sécurité inhérents à l'IoT, il est impératif d'assurer la sécurité des systèmes IoT. Par conséquent, il est nécessaire de mettre en œuvre un système sécurisé pour l'Internet des objets, les normes et standards ont fixé des exigences de sécurité qui garantissent la sécurité de l'information et du réseau. Nous présentons ci-dessous un résumé de ces exigences citées dans la littérature [8].

- **La confidentialité :** La confidentialité peut être un gros problème pour les systèmes IoT, surtout lors de la transmission des données confidentielles, ces informations sont sensibles et ne doivent pas être divulguées à des lecteurs non autorisés utilisant des étiquettes RFID électroniques. La confidentialité peut être obtenue en faisant confiance aux tiers pour qu'ils n'utilisent pas de manière abusive les données générées par les systèmes de l'IdO, ou bien de contrôler rigoureusement la collecte et l'utilisation de ces données. [8].

- **L'ntégrité :** La protection de l'intégrité garantit que les données transmises ne sont pas fabriquées, c'est-à-dire non réécrites, copiées ou remplacées par l'attaquant et

que seules les parties autorisées peuvent les modifier ou supprimer. L'intégrité du système IoT ne se limite pas à protéger les appareils IoT contre la falsification, elle vise également à détecter les tentatives de falsification sur ces appareils [8].

- **La disponibilité :** Dans le contexte de l'IoT, la disponibilité nécessite l'utilisation de divers services en temps voulu fournis par IOT, en mettant à jour tout le matériel et les logiciels pour empêcher les attaques DOS qui visent cette exigence.[8].

- **L'authentification :** Dans le contexte de l'authentification, il s'agit de vérifier l'identité d'un utilisateur, d'un processus ou d'un appareil. Tout dispositif IoT doit pouvoir justifier de son identité afin d'obtenir certains droits d'exploitation ou d'accès aux ressources du système d'information. Généralement, c'est l'attaque d'usurpation d'identité qui porte atteinte à cette exigence. [24].

- **l'autorisation :** Dans le contexte de l'IoT, l'autorisation peut être définie comme le droit ou la permission accordée à une entité système d'accéder à une ressource système, tel que l'accès à des ressources telles que des données de capteurs ou un fichier. L'autorisation limite les privilèges des appareils, des utilisateurs, des applications et des composants, afin qu'ils ne puissent accéder qu'aux ressources nécessaires à leur fonctionnement normal [24].

- **La non-répudiation :** La non-répudiation peut être définie comme l'empêchement de tout utilisateur ou appareil de nier une action, en fournissant des preuves de ces actions. La non-répudiation n'est généralement pas considérée comme une exigence de sécurité essentielle pour de nombreuses applications IoT, mais elle devient d'une importance critique dans le contexte des applications commerciales [24].

.

1.13 Avantages et inconvénients du réseau IoT

L'Internet des objets (IoT) offre une connectivité sans précédent entre les appareils, ouvrant ainsi la voie à de nombreuses opportunités, mais il présente également des défis et des considérations à prendre en compte. Nous allons les voir Dans ce qui suit.

1.13.1 Avantages :

- Automatisation : L'IoT permet l'automatisation de nombreuses tâches, ce qui peut faciliter la vie quotidienne et améliorer l'efficacité.

- Surveillance : L'IoT permet de surveiller à distance des objets et des environnements. Par exemple, cela peut être utile pour surveiller des installations industrielles, des réseaux de distribution d'eau, des stations météorologiques, etc.

- Précision : L'IoT peut améliorer la précision des processus. Par exemple dans l'industrie.

- Innovation : L'IoT offre des possibilités d'innovation dans de nombreux domaines, notamment la santé, l'agriculture, l'industrie, l'énergie, le transport, etc.

1.13.2 Inconvénients :

- Sécurité : Les risques de sécurité constituent une préoccupation importante pour l'IoT, car la connexion de nombreux objets à Internet les expose potentiellement à des cyberattaques.
- Vie privée : L'IoT peut collecter de grandes quantités de données personnelles, ce qui peut poser des problèmes de protection de la vie privée. Les données pouvant être utilisées pour suivre les comportements des utilisateurs, ce qui peut être considéré comme intrusif.
- La mise en place et la maintenance de l'IoT nécessitent des compétences spécialisées en raison de sa complexité technologique.
- Coûts : La mise en place de l'IoT peut être nécessaire en raison de la nécessité d'acheter des capteurs et des dispositifs connectés, ainsi que de la mise en place de l'infrastructure nécessaire.

Pour conclure, bien que l'IoT présente de nombreux avantages, il comporte également des défis qui nécessitent une évaluation au cas par cas en fonction des besoins de chaque utilisateur et du contexte d'utilisation. Il est crucial d'examiner attentivement les avantages et les inconvénients avant de décider d'adopter cette technologie, et de prendre des mesures pour réduire les risques potentiels qui y sont associés.

S'assurer que toutes les exigences de sécurité dans un réseau IoT contribuent à renforcer la sécurité et à réduire le risque d'attaques, c'est pourquoi il est impératif de respecter ces exigences dans les communications de bout en bout.

1.14 Conclusion

En conclusion, l'Internet des objets (IoT) représente une avancée majeure dans le domaine de la connectivité et de la technologie et offre un potentiel énorme pour améliorer notre vie quotidienne. Au cours de ce chapitre, nous avons examiné les généralités de l'IoT, en comprenant ses composants essentiels, ses applications diverses et ses implications pour l'avenir. Dans ce qui suit, nous allons approfondir notre compréhension de la sécurité dans les réseaux IoT utilisant le protocole RPL, ainsi que des différentes attaques auxquelles ils sont susceptibles d'être confrontés.

Chapitre 2

Étude des attaques RPL

2.1 Introduction

Dans le monde de l'Internet des objets (IoT), la connectivité et la communication sécurisée sont des aspects essentiels pour assurer le bon fonctionnement des systèmes et des dispositifs interconnectés. Le protocole Routing Protocol for Low-power and Lossy Networks (RPL) est l'un des protocoles les plus couramment utilisés dans les réseaux de capteurs sans fil et l'IoT. Dans ce chapitre, nous allons examiner les généralités sur le protocole RPL ainsi que les attaques visant spécifiquement ce protocole .

2.2 les réseaux LLN (Low Power and Lossy Networks)

LLN fait partie de l'Internet des objets. IL se compose généralement d'un grand nombre d'appareils intégrés interconnectés via différents types de connexions, telles que : IEEE 802.15.4, 6LOWPAN, Zigbee, Bluetooth Low Power, Wifi Low Power, etc. Un réseau LLN se compose de nœuds avec une puissance, une mémoire et une puissance de traitement limitées, de sorte que les connexions des nœuds au sein d'un LLN présentent une perte de paquets élevée et de faibles débits de données. Ils se composent de dizaines de milliers de routeurs LLN, prenant en charge le trafic point à point, le trafic point à multipoint et le trafic multipoint à point. Un réseau de capteurs sans fil (WSN) est un type de LLN où les nœuds collectent les données des capteurs et les envoient à une station de base ou à un puits.
Les LLN ont de nombreuses applications dans divers domaines tels que la surveillance industrielle, l'automatisation des bâtiments (l'éclairage, l'accès et la protection incendie), la domotique, la santé, la surveillance environnementale, les réseaux de capteurs urbains, la gestion de l'énergie, le suivi des actifs et le refroidissement. La fonctionnalité qui décrit le protocole de routage IPv6 pour les LLN (RPL) est connue sous le nom de protocole de routage pour les réseaux à faible puissance et avec perte.

2.3 Routage dans les LLNs

Des tests ont été menés par l'IETF sur les protocoles de routage qui sont actuellement spécifiés dans les RFC, les chercheurs ont conclu qu'aucun d'entre eux n'était suffisant pour répondre aux normes requises pour les LLN. Cela a démontré un besoin évident d'un protocole de routage conçu explicitement pour répondre aux contraintes de routage dans les LLN, ce qui a conduit à la création du protocole de routage RPL. l'IETF (Internet Engineering Task Force) que les protocoles de communication pour les réseaux à faible puissance et avec perte (LLN) doivent fonctionner avec de fortes contraintes. Ces contraintes incluent des ressources énergétiques limitées, une faible bande passante et des taux de perte de paquets élevés. Le groupe ROLL de l'IETF a établi une norme pour ces réseaux afin de s'assurer que leurs protocoles de communication sont optimisés pour ces limitations. Sous la référence RFC6550, l'IETF (Internet Engineering Task Force) opère.

2.4 Le protocole de routage RPL

Le protocole RPL(Routing Protocol for Low power and Lossy Networks), est un protocole de routage proactif, et il est le standard recommandé par l'IETF pour les réseaux basse consommation et à faible puissance (LLN) et les réseaux de capteurs 6LoWPAN, il fonctionne sur la norme(IEEE 802.15.4). Il a été mis à jour en mars 2012. RPL offre une connectivité IPv6 à Internet et réduit également le coût pour atteindre la station de base à partir de n'importe quel nœud du LLN. Le protocole RPL est principalement destiné aux réseaux de collecte, où les nœuds envoient régulièrement des mesures à un point de collecte. Il a été spécialement conçu pour s'adapter aux conditions changeantes du réseau et pour fournir des itinéraires de secours en cas d'indisponibilité des itinéraires par défaut [55].

2.4.1 La topologie RPL

Le protocole RPL établit une structure de routage appelée DODAG(Directed Acyclic Graph) qui est enracinée au niveau de la passerelle. Le DODAG est construit selon un processus de découverte de voisins (Neighbor Dis-covery (ND)). C'est un arbre de routage créé par un nœud racine, il décrit les liens orientés entre les nœuds, se terminant à un ou plusieurs nœuds racines, ou chaque nœud peut transmettre des données à son nœud parent, qui les transmet à son tour vers le haut jusqu'à ce qu'elles atteignent le nœud de destination ou la passerelle. De même, le nœud de destination peut envoyer un message unicast pour cibler un nœud spécifique dans son réseau [31].

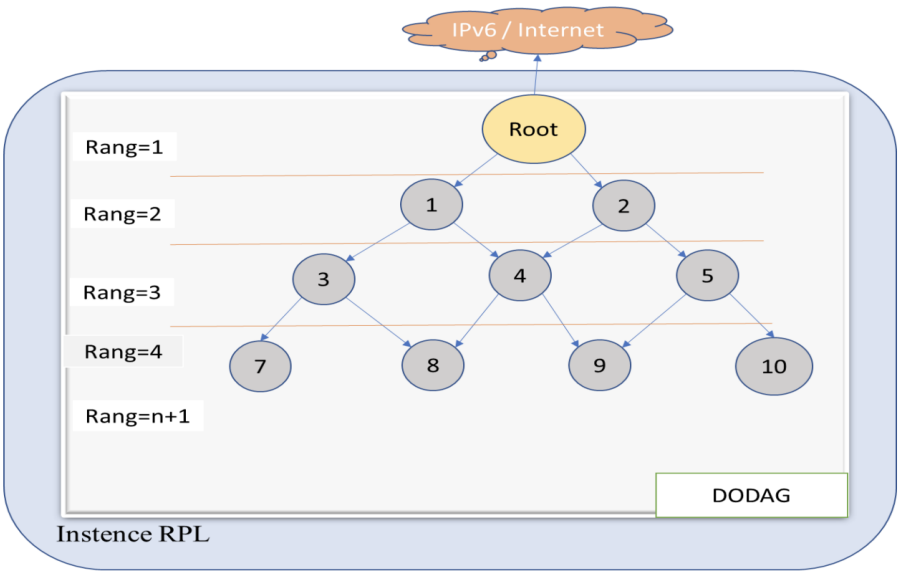

Fig. 2.1 : Partition de topologie RPL [28].

2.4.2 Les messages de controle dans RPL

Afin de maintenir la topologie de routage, le protocole RPL utilise quatre nouveaux messages de contrôle ICMPv6.

Le DIO

Le premier message est appelé DIO (DODAG Information Object), il est utilisé pour créer des routes ascendantes pour permettre à un noeud de découvrir une instance RPL et de la rejoindre. Seuls la racine RPL fournit les paramètres nécessaires pour mettre en place la topologie, tandis que les autres nœuds ne servent que de relais. Les DIO se propagent dans le réseau en sens inverse et sont diffusés par défaut en multidiffusion, mais peuvent également être diffusés en monodiffusion sur une demande spécifique d'un nœud.

Le DIS

Le deuxième message est appelé DIS (DODAG Information Sollicitation), il est utilisé par un nœud pour rejoindre la topologie (diffusion en multidiffusion) pour demander des informations de configuration plus récentes sur le DODAG.Tout nœud recevant un DIS répond à l'initiateur par une diffusion en monodiffusion d'un paquet DIO. En d'autre terme il demande un message DIO.

Le DAO

Le troisième message est DAO (DODAG Advertisement Object), il est utilisé uniquement si des routes descendantes sont nécessaires (par exemple, pour le trafic point à point). Sinon, il peut être désactivé pour économiser les ressources. En mode de fonctionnement non conservation (non-storing mode), le DAO est envoyé à la racine RPL, tandis qu'en mode conservation (storing mode), il est envoyé aux parents. Contrairement aux autres messages de contrôle RPL, le DAO est toujours envoyé en monodiffusion et nécessite un accusé de réception du destinataire.

Le DAO-Ack

Le quatrième message est le DAO-Ack (DAO-Acknowledgment), il est utilisé par le noeud parent au noeud fils, en réponse à son message DAO reçu pour confirmer la réception. Si la source ne reçoit pas de DAO-Ack, elle peut réémettre le DAO initial [31].

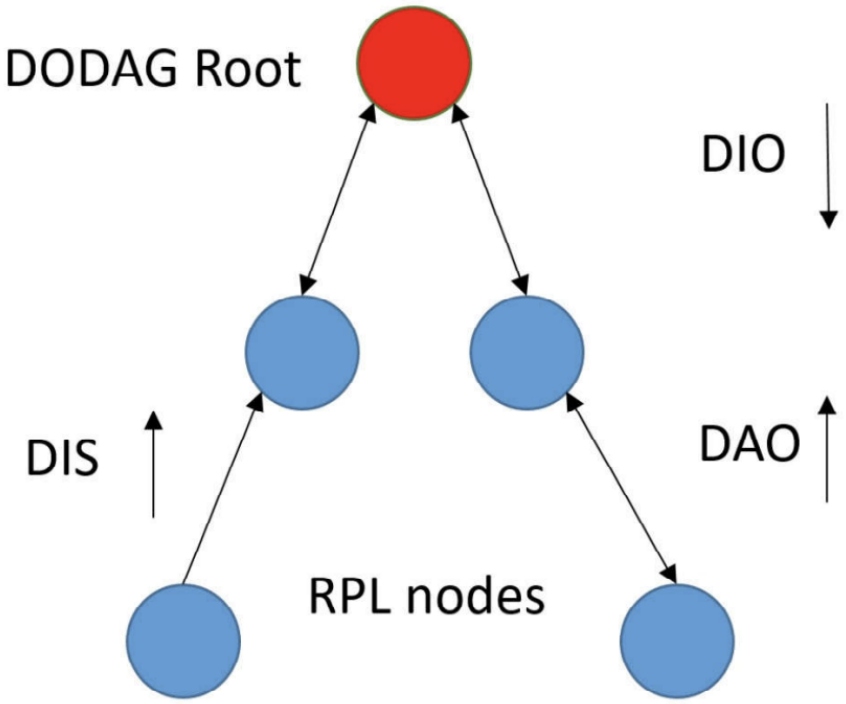

Fig. 2.2 : Control messages in RPL [31].

2.4.3 La Construction du DODAG

La construction du DODAG s'appuie sur le processus du protocole Neighbour Discovery (ND) utilisé avec l'IPv6. Le graphe DODAG est construit étape par étape :

- **Etape 1:** Le processus de construction de DODAG est initié par la racine du DODAG, qui diffuse périodiquement un message DIO à tous ses nœuds voisins. Ce message transmet des paramètres importants comme un DODAGID, sa fonction Objectif, ainsi que des informations pour permettre aux nœuds de déterminer leur rang dans le DODAG, pour créer et maintenir le graphe DODAG. La valeur de rang d'un nœud correspond à sa position dans le graphe par rapport à la racine et doit toujours être supérieure au rang de ses parents.
- **Etape 2:** Chaque fois qu'un nœud RPL reçoit un message DIO, il doit d'abord décider s'il l'accepte ou non, S'il ne répond pas à certains critères définis par RPL, il sera rejeté. Sinon, le nœud procède au traitement du message DIO.
- **Etape 3:** Une fois qu'un nœud reçoit un message DIO pour la première fois et décide de rejoindre le DODAG, il ajoute l'adresse de l'émetteur DIO à sa liste de parents et calcule son rang. Il transmet ensuite le message DIO avec les informations de mise à jour du rang à ses voisins. Sur la base de sa liste de parents, le nœud sélectionne un parent préféré qui devient la passerelle par défaut à utiliser lorsque des données doivent être envoyées vers la racine du DODAG.
- **Etape 4:** lorsqu'un noeud est déjà associé à un DODAG et qu'il reçoit un autre message DIO, il calcule son nouveau rang et le compare à l'ancien. Si le nouveau

rang est inférieur, le nœud ajoute l'expéditeur du message à sa liste de parents et le choisit comme nouveau parent préféré. Ensuite, le nœud met à jour le message DIO avec le nouveau rang et le diffuse à ses voisins, sinon si le nouveau rang calculé est supérieur à l'ancien, le nœud ne met pas à jour le rang ni n'envoie de message DIO. Il conserve le parent préféré précédent.

- **Etape 5:** Un noeud qui n'a reçu aucun message DIO et n'est pas associé à un DODAG peut demander des informations sur le réseau en envoyant périodiquement des messages DIS aux voisins.
- **Etape 6:** À la fin de ce processus, tous les nœuds participants au graphe DODAG disposent d'une route par défaut ascendant vers la racine du DODAG [40].

2.4.4 les modes d'opération du RPL

RPL a deux modes de fonctionnement pour le routage de liaison descendante :

- **Mode Storing :** Dans ce mode, les nœuds intermédiaires ont la capacité de stocker des informations de routage afin de pouvoir rediriger les données reçues vers la destination appropriée [35].
- **Mode Non-Storing :** Tous les messages descendants doivent d'abord passer par le nœud racine, puisque seule la racine possède des informations de routage descendantes [35].

2.4.5 Modes de communication

RPL supporte trois modes de communication :

- **Mode multipoint à point (MP2P) :** RPL est principalement conçu pour optimiser la communication multipoint à point (MP2P) des flux de trafic. Les nœuds clients peuvent envoyer des messages DAO à la racine DODAG via la voie montante en utilisant la communication MP2P. Cependant, la condition préalable est que le nœud soit connecté au réseau. Le protocole se concentre sur l'amélioration de la qualité de service (QoS) des flux de trafic MP2P en utilisant des mesures de coût basées sur l'énergie et le temps de transit. Par conséquent, RPL peut fournir une communication fiable et efficace pour les applications MP2P telles que la surveillance environnementale et la collecte de données à distance [25].

- **Le mode point à multipoint (P2MP) :** RPL prend en charge le trafic P2MP et utilise un mécanisme de publicité de destination pour créer des itinéraires descendants à partir de la racine vers d'autres nœuds. Cette fonctionnalité est utilisée pour transférer des données vers un préfixe, une adresse ou un groupe de multidiffusion. Les messages DIO et P2PM sont utilisés dans plusieurs applications de réseau de capteurs sans fil (LLN) qui nécessitent ce type de trafic. Ce service de publicité

permet de fournir des routes descendantes d'une racine vers une ou plusieurs destinations à l'aide d'une adresse de multidiffusion [25].

- **Le mode point à point (P2P) :** avec le protocole RPL, le trafic P2P peut être acheminé via deux modes : le mode Storing et le mode Non-Storing. Dans le mode Storing, le paquet passe par un ancêtre pour atteindre la destination, qui peut être soit l'ancêtre lui-même, soit un nœud plus proche de la source ou de la destination. En revanche, dans le mode Non-Storing, le paquet est dirigé vers la racine, qui est le point de passage obligatoire pour tous les paquets avant d'être acheminés vers la destination [25].

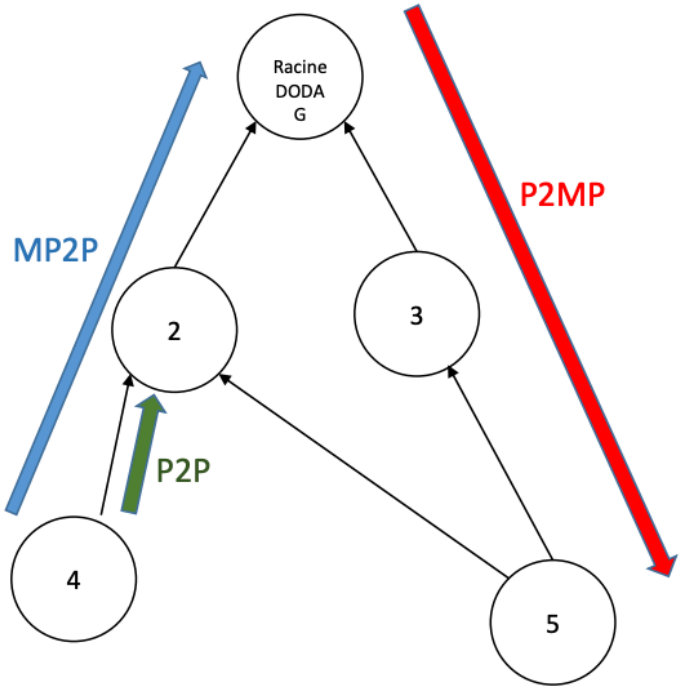

FIG. 2.3 : Modèles de communication [36].

2.4.6 Le rang (rank)

Chaque nœud dans un DODAG a un rang qui indique la position d'un nœud par rapport aux autres nœuds et par rapport au DODAGracine. Les rangs décroissent strictement dans le sens ascendant vers lA racine DODAG et augmentation strictement à partir de la racine DODAG vers les nœuds. La fonction d'objectif OF est responsable du calcul de cette valeur et elle doit toujours être supérieure au rang de ses parents pour éviter toute boucle dans le graphe [20].

2.4.7 La fonction objectif (OF)

Dans le RPL DODAG, une fonction objectif (OF) est utilisée pour définir des paramètres importants tels que les métriques de routage, les objectifs d'optimisation, la formule de calcul du rang et les critères de sélection du parent. L'IETF a défini deux fonctions d'objectifs pour RPL : OF0 et MRHOF.

- **Fonction Objectif Zéro (OF0) :** cette fonction utilise le nombre de sauts comme métrique. Ainsi, le rang d'un nœud calculé selon OF0 correspond essentiellement au nombre de sauts entre le nœud et la racine. OF0 est la fonction objectif par défaut pour faciliter l'interopérabilité entre différentes implémentations du protocole RPL [30].
- **Fonction objectif Minimum Rank with Hysteresis (MRHOF) :** utilise l'hystérésis pour choisir le chemin avec la métrique la plus petite. La métrique utilisée par MRHOF est déterminée par le champ "metric container" des messages DIO. Par défaut, cette fonction utilise le compteur de transmission attendu (ETX) pour calculer le rang. L'ETX d'un lien est défini comme le nombre de transmissions attendues nécessaires pour transférer avec succès un seul paquet sur un lien [30].

2.4.8 Les types de nœud dans RPL

En effet, le protocole RPL définit trois types de nœuds qui peuvent être présents dans un réseau :

1. **Les LBR (Low Power and Lossy Border Routers) :** ce sont les nœuds qui font office de racine (root) d'un DODAG et ont la capacité de construire et de maintenir la topologie du réseau. Ils agissent également comme passerelle pour le trafic entre le réseau de capteurs et le reste d'Internet.
2. **Les routeurs :** ce sont des nœuds qui peuvent transférer et générer du trafic, mais qui ne peuvent pas créer un nouveau DODAG. Ils se connectent à un DODAG existant et servent de relais pour les paquets.
3. **Les hôtes :** ce sont des périphériques finaux qui peuvent générer du trafic de données, mais qui ne sont pas en mesure de transférer le trafic. Ils sont généralement équipés de capteurs et collectent des données pour les envoyer à des nœuds routants [25].

2.4.9 Maintenance de la topologie

La réparation globale et locale sont deux approches utilisées dans le protocole RPL pour résoudre les problèmes de connectivité et maintenir la topologie du réseau.

- **Réparation globale :** intervient lorsque la connectivité avec la racine est perdue. Elle implique la diffusion de messages de contrôle (DIO) à travers le réseau pour

informer les nœuds de la perte de connectivité et pour initier la recherche d'une nouvelle route. Réparation couteuse en trafic.

- **La réparation locale :** survient lorsque la connexion avec le parent direct d'un nœud est perdue. Dans de tels cas, le nœud affecté peut essayer de trouver un nouveau parent pour maintenir sa connectivité. Cela implique généralement l'envoi de messages de contrôle, tels que des messages DAO. L'objectif de la réparation locale est de rétablir la connectivité pour le nœud individuel sans perturber la topologie globale du réseau.

2.4.10 Trickle Timer

RPL utilise l'algorithme du "Trickle timer" pour réduire la charge des messages de contrôle dans le réseau. Cet algorithme se déroule comme suit [20] :

1. Définir les paramètres du "trickle timer" :
   - MinInterval : l'intervalle de temps minimum entre les envois de messages.
   - MaxInterval : l'intervalle de temps maximum entre les envois de messages.
   - I : une variable d'ajustement initiale.
   - K : un facteur d'ajustement.

2. Initialisation :
   - timer := MinInterval
   - counter := 0

3. Tant que le routeur est actif :
   - Si le timer s'est écoulé :
   - Envoyer un message de mise à jour d'état.
   - Réinitialiser le timer à MinInterval.
   - Incrémenter le compteur.
   - Si le compteur atteint une valeur prédéfinie (par exemple, 2 fois), réinitialiser le compteur et doubler la valeur du timer.
   - Sinon, si le compteur dépasse un seuil maximal (par exemple, 6 fois), réinitialiser le compteur et diviser la valeur du timer par K.

   - Attendre un certain intervalle de temps (par exemple, 1 seconde) avant de vérifier à nouveau le timer.

2.5 Attaques sur le protocole RPL

Le protocole RPL est exposé à diverses attaques de sécurité lors du transfert de paquets de données entre les appareils. Les caractéristiques des réseaux LLN, telles que les ressources limitées, le manque d'infrastructure, la sécurité physique limitée, les topologies dynamiques et les liaisons peu fiables, les rendent particulièrement vulnérables et difficiles à défendre contre les attaques. Ça peut-être bien que spécifique au protocole RPL, il s'applique également aux réseaux de capteurs sans fil, et aux réseaux filaires [40].
Dans ce qui suit nous présenterons les attaques courantes sur RPL.

2.5.1 Attaques contre les ressources

Attaque DIS

Des messages de contrôle liés au RPL sont transmis dans le réseau pour construire une structure de transmission optimisée. Les nœuds malveillants internes peuvent attaquer le réseau RPL en envoyant un grand nombre de messages de contrôle inutiles. L'une de ces attaques cible les messages de contrôle DIS envoyés par de nouveaux nœuds pour rejoindre le réseau. Cette attaque est appelée attaque DIS.

Lorsqu'un nouveau nœud souhaite rejoindre le réseau, il envoie périodiquement des messages de contrôle DIS pour demander des informations d'autorisation. Les nœuds malveillants peuvent exploiter ce processus en inondant le réseau d'un grand nombre de messages DIS, ce qui affecte négativement les performances du réseau. Cette attaque est conçue pour perturber le fonctionnement normal des protocoles de routage en augmentant le trafic inutile et en épuisant les ressources du réseau. Les conséquences d'une telle attaque pourraient inclure une congestion du réseau, une consommation d'énergie accrue et une qualité de service réduite [10].

Version number attack

Le numéro de version est un champ important de chaque message DIO. Il est propagé sur le graphe DODAG et est incrémenté par la racine seulement, chaque fois, la reconstruction du DODAG est nécessaire, Dans cette attaque, l'attaquant augmente le champ du numéro de version dans les messages DIO et les transmet à ses voisins. En conséquence, la reconstruction inutile d'un nouveau DODAG est forcée, ce qui entraîne la perte de paquets de données, l'encombrement du réseau et l'épuisement des ressources des nœuds en raison de la surcharge des messages de contrôle [40].

Attaque DOS

Un attaquant pourrait essayer d'envoyer de nombreux messages pour brouiller un canal réseau. Cela réduit les performances et l'efficacité du réseau. Un attaquant extérieur pourrait lancer une attaque par déni de service (DoS) en envoyant des messages invalides sur le réseau et en exfiltrant les messages des nœuds légitimes. Cela empêche les nœuds

légitimes de traiter les messages en raison de messages non-valides provenant d'attaquants. Par rapport aux attaques DoS, les attaques par déni de service distribué (DDoS) sont des attaques plus graves dans lesquelles un groupe de nœuds voyous lancent des attaques contre des nœuds légitimes à différents endroits et à différents moments. Ce type d'attaque consomme des ressources, réduit la capacité du réseau et empêche le réseau de fonctionner correctement ou en temps opportun. [37].

2.5.2 Attaques contre la topologie

Attaque Sinkhole

Dans une attaque sinkhole, un nœud compromis achemine intentionnellement tout le trafic provenant de sa zone voisine vers lui-même en diffusant de fausses informations, un attaquant ou un nœud compromis essaie d'inciter d'autres nœuds à envoyer des données en se faisant passer pour les relais les plus attrayants de leur voisinage. Le but de cette attaque est d'intercepter et de manipuler les données lorsqu'elles traversent le réseau [37].

Attaque Rank

L'un des principaux éléments de conception de RPL est son mécanisme de classement, qui utilise la propriété de classement pour assurer un routage sans boucle. Un nœud peut changer sa valeur de classement en manipulant les valeurs de rang() des nœuds de manière malveillante et tromper ses nœuds voisins après avoir rejoint un réseau RPL. L'objectif de cette attaque est de créer une topologie sous-optimale, entraînant un trafic de données empruntant des chemins réseau de moindre qualité de service (QoS). Plusieurs nœuds adjacents changeront ultérieurement leur parent préféré actuel. Cela peut alors déclencher plus facilement d'autres attaques, comme les trous noirs et aggraver les choses. L'objectif principal des deux modes d'attaque de classement est de déstabiliser le réseau [10].

Attaque Wormhole

Une attaque par trou de ver est une attaque grave qui peut être lancée même lorsque l'authenticité et la confidentialité sont garanties dans toutes les communications. Une attaque par trou de ver est une attaque dans laquelle deux ou plusieurs nœuds attaquants sont connectés par un lien appelé lien de trou de ver, et les nœuds forment un tunnel pour diffuser des paquets de données dans le réseau.
Dans le cas d'un réseau sans fil, il est plus facile d'effectuer cette attaque, car un attaquant peut envoyer à travers le trou de ver le trafic qui lui est envoyé ainsi que tout trafic intercepté lors de la transmission sans fil. L'attaque par trou de ver déforme le chemin de routage et est particulièrement problématique pour les réseaux RPL. Si un attaquant transmet des informations à une autre partie du réseau, les nœuds qui sont en fait éloignés se verront comme s'ils étaient dans le même voisinage. En conséquence, ils peuvent générer des routes non optimisées vers la fonction objective. Cela embrouille le réseau et perturbe le processus de communication. [40].

Attaque Blackhole

Dans cette attaque, le nœud attaquant prétend qu'il a le chemin le plus court vers la destination le nœud de contrôle (appelé node sink), après avoir illégalement changé son rang. Dans une attaque par trou noir, un intrus malveillant sa seule mission est alors de ne rien transférer, créant une sorte de puits ou de blackhole dans le réseau. Laisse tomber tous les paquets qu'il est censé transmettre. Cette attaque peut être très préjudiciable lorsqu'elle est combinée à une attaque de gouffre [53], entraînant la perte d'une grande partie du trafic. Elle peut être considérée comme un type d'attaque par déni de service. Si l'attaquant occupe une position stratégique dans le graphe, il peut isoler plusieurs nœuds du réseau. Il existe également une variante de cette attaque appelée trou gris (ou attaque par transfert sélectif) dans laquelle l'attaquant ne rejette qu'une partie spécifique du trafic du réseau [40]. Les nœuds doivent communiquer de manière adéquate pour former un DODAG légal sans problème. En outre, lors du lancement d'une attaque par trou noir, le nœud malveillant ne génère aucun message de contrôle [32]. Une attaque par trou noir peut être orchestrée par un seul nœud malveillant ou par un groupe de nœuds malveillants qui s'entendent pour rendre l'attaque plus difficile à détecter [43].

2.5.3 Attaques contre le trafic

Attaque Sybil

Sybil est également appelée l'attaque « nœud unique avec plusieurs identités ». Le nœud malveillant affiche un identifiant différent et peut se trouver à plusieurs endroits en même temps et il ressemble à un cœur ordinaire. Ce dernier peut exploiter le mécanisme de transmission DIS pour attaquer également le réseau. Si le nœud malveillant génère et multidiffuse un grand nombre de messages DIS superposés avec différentes identités fictives, tous les nœuds récepteurs vont croire que de nouveaux nœuds veulent rejoindre le réseau, puis redémarrer l'algorithme Trickle depuis le début à plusieurs reprises et diffuser un nombre excessif de messages DIO. Cette attaque dégrade les performances du système. [42].

Attaque analyse du trafic

permettent d'obtenir des informations sur le routage en analysant les schémas de trafic d'une liaison, même si les paquets sont chiffrés. L'objectif est de collecter des informations sur le réseau RPL, telles qu'une vue partielle de la topologie en identifiant les relations entre les nœuds parents et enfants. Un nœud malveillant peut ensuite utiliser ces informations pour mener d'autres attaques. Si l'attaquant est proche du nœud racine, il peut traiter un volume de trafic plus important et obtenir davantage d'informations que s'il se trouve en périphérie d'un sous-DODAG [40].

2.6 Contre-mesure

Plusieurs mesures peuvent être envisagées pour protéger les réseaux RPL contre des attaques. Des mécanismes peuvent être utilisés pour assurer la fiabilité des données échangées, de l'intégrité des données, de l'authentification des nœuds (node ID), etc. Les solutions de sécurité et les mécanismes d'atténuation peuvent être regroupés en deux grandes catégories :

- Mécanismes des IDS : La première catégorie comprend l'utilisation de systèmes de détection d'intrusion (IDS). Ces méthodes sont conçues pour détecter différentes attaques et réduire leurs impacts sur le réseau. Les solutions IDS surveillent en permanence le réseau à la recherche d'activités suspectes. Lorsqu'une attaque est détectée, elles tentent d'identifier l'attaquant et de limiter les conséquences de l'attaque sur le réseau. Les IDS peuvent être catégorisés selon leur emplacement (Centralisé, Distribué, Hybride) ou la méthode de détection (Basé sur la Signature, Basé sur les Anomalies, Basé sur les Spécifications, Basé sur l' Hybride, Basé sur les Événements). de nombreuses études IDS sur ce sujet, dont [54].

- Les solutions de la deuxième catégorie visent à contrer des attaques spécifiques en introduisant des mécanismes supplémentaires dans RPL. Cela peut inclure l'ajout de nouveaux messages de contrôle, la modification de certains paramètres, l'établissement de seuils ou l'utilisation de solutions cryptographiques en chiffrant les données . L'objectif est de prévenir l'attaque ou de limiter son impact sur le réseau pour assurer la confidentialité des informations échangées entre les nœuds du réseau. Cela empêche les attaquants d'intercepter et de lire des données sensibles [39].

2.7 Les métriques du RPL

Différents paramètres ayant un impact sur le processus de routage seront pris en compte pour comparer les performances de RPL dans les réseaux statiques et mobiles[35]. ces différentes métriques sont définies comme suit :

2.7.1 Consommation d'énergie

La consommation d'énergie, qui se rapporte directement à la durée de vie d'un nœud, représente la quantité d'énergie que ce dernier consomme. La consommation d'énergie est étroitement liée au nombre de messages transmis et reçus par le nœud, au temps de traitement et au risque de surchauffe en cas d'inactivité.

2.7.2 ETX (Expected Transmission Count)

L'ETX (expected transmission count) correspond au nombre maximal de retransmissions nécessaires pour qu'un paquet soit correctement acheminé vers sa destination. Cet

indicateur reflète la performance de la RPL. En cas de mauvaise qualité de la liaison, l'ETX augmente, car moins de paquets parviennent à leur destination, nécessitant davantage de retransmissions.

2.7.3 PDR(Packet Delivery Ratio)

Le taux de distribution des paquets (PDR) mesuré par le nombre de paquets délivrés à la destination avec succès, en comparant avec le nombre de paquets envoyés par l'émetteur. Avoir un PDR élevé indique une performance supérieure de la RPL.

2.7.4 End to end delay

Le temps nécessaire pour qu'un paquet se déplace de sa source à sa destination est appelé délai de bout en bout. Cette mesure est cruciale pour les scénarios sensibles aux délais. Le délai de bout en bout fournit également des informations sur la qualité du réseau et sur les liens utilisés. Si le délai est élevé, cela peut indiquer une défaillance RPL, une mauvaise performance des liens, ou des deux.

2.7.5 Les messages de contrôle

Le trafic aérien représente la quantité de messages de contrôle RPL envoyés par les nœuds, tels que les messages DIO, DAO et DIS. La quantité de messages de contrôle a un impact direct sur la consommation d'énergie dans un réseau, ce qui rend l'optimisation du processus de contrôle cruciale.

2.8 Conclusion

Ce chapitre a examiné en détail les attaques ciblant spécifiquement le protocole RPL dans les réseaux IoT. Nous avons exploré les vulnérabilités et les risques associés à l'utilisation du protocole RPL, ainsi que les conséquences potentielles de ces attaques sur la sécurité des réseaux.

Chapitre 3

État de l'art

3.1 Travaux connexes

Dans le cadre de l'état de l'art, l'attention particulière est portée sur les attaques liées à l'Internet des objets (IoT), et plus précisément sur les attaques de routage utilisant le protocole RPL (Routing Protocol for Low-Power and Lossy Networks). Afin de représenter clairement les résultats de notre recherche, nous avons élaboré le diagramme schématique ci-dessous qui résume les différentes étapes et les liens entre les attaques de routage identifiées.

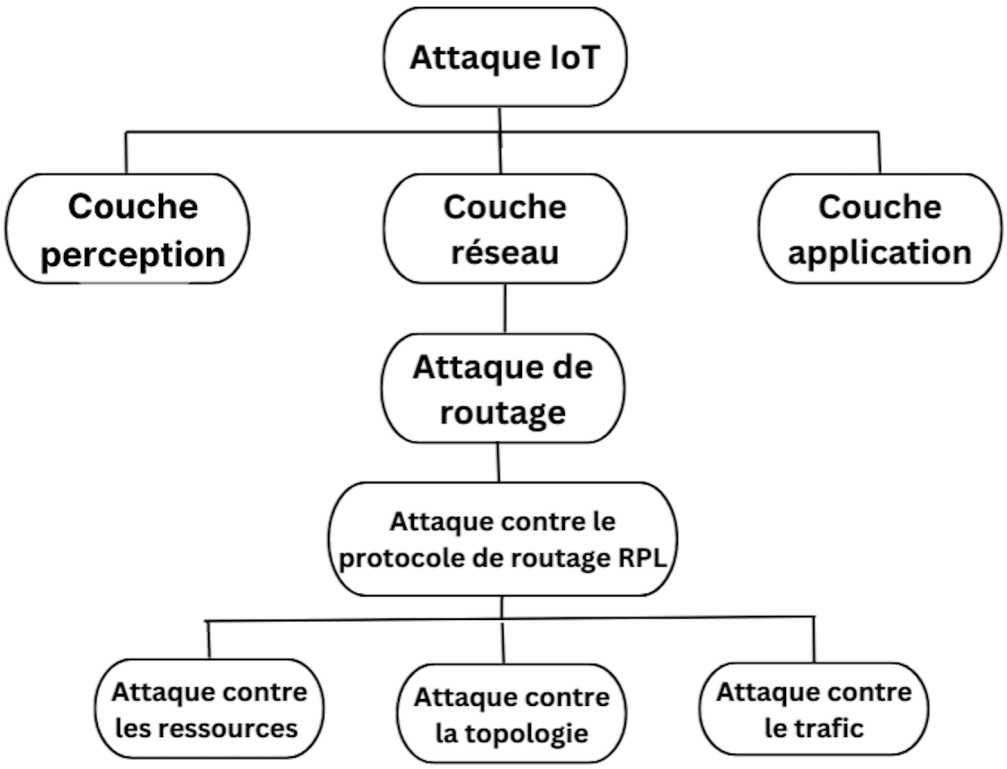

Fig. 3.1 : Méthodologie de l'état de l'art

3.1.1 Attaques dans les systèmes IoT

Andrea et al. [11], Arshad et al. [29] et Lin et al. [51] ont catégorisé les attaques IoT en fonction des niveaux sur lesquels elles agissent, telles que les attaques physiques, logicielles, réseau, de chiffrement et de routage. Les auteurs ont exclu les attaques environnementales de leur taxonomie, considérant qu'un attaquant n'a aucun contrôle sur ces phénomènes. Cependant, une attaque peut appartenir à plusieurs catégories, comme c'est le cas pour les attaques par déni de service (DoS).

Younes ABBASSI et al. [8] abordent la problématique de la sécurité dans l'IoT, en premier lieu ils présentent les 3 couches principales de l'iot et mettent en évidence les

différentes vulnérabilités potentielles de l'IoT telles que l'absence de cryptage des transports, authentification et autorisation insuffisantes, interface Web non sécurisée, logiciels et micrologiciels non sécurisés, attaques numériques. Ensuite, ils présentent les principales solutions de sécurité existantes, telles que l'authentification, la confidentialité, l'intégrité et la disponibilité, ainsi que les approches émergentes de sécurité pour l'IoT.

Mohamed Litoussi et al. [26] ont traité la sécurité de l'Internet des objets (IoT) et des défis qui y sont associés. L'IoT est un réseau de dispositifs physiques interconnectés, tels que des capteurs, des caméras et des dispositifs de contrôle, qui peuvent communiquer et échanger des données via Internet. Cependant, la sécurité de ces dispositifs est souvent mise en danger en raison de leur connectivité Internet, de leur manque de mises à jour de sécurité régulières.

Ils ont articulé les trois couches essentielles de l'architecture de l'IdO et les différents types d'attaques sur ces couches avec des exemples. Dont perception layer, network layer et application layer ; les auteurs ont classifié les menaces de sécurité courantes comme suit :

Perception layer	Network layer	Application layer
-Eavesdropping. -Replay Attack. -Timing Attack	-Denial of service (DoS attack). -RFID spoofing. -Sinkhole attack	-Phishing attack. -Cross site scripting. -Malicious virus-worm.

Tab. 3.1 : classification des attaques selon [26] .

Butun et al. [48] ont présenté le concept d'attaques passives et actives et les ont classées en cinq couches (physique, MAC, réseau, transport et application). La sécurité des dispositifs IoT peut être compromise directement ou indirectement à travers ses différents composants tels que les réseaux de capteurs sans fil (WSN), le cloud, les analyses, l'interface utilisateur, les passerelles ou les dispositifs de l'utilisateur final. Les cybercriminels utilisent des méthodes d'attaques intelligentes, brutales, intelligentes et furtives qui réduisent la probabilité d'être détectées.

Classification des attaques

1. **Couche physique (Physical Layer) :** La couche physique IoT pose plusieurs défis, notamment en termes de sécurité, de fiabilité et de compatibilité. De plus, les appareils IoT peuvent être compromis par plusieurs types d'attaques cela peut avoir des conséquences graves sur la sécurité des données et la vie privée des utilisateurs.

-Le tableau 3.2 analyse brièvement les attaques de la couche physique :

Nom de l'attaque	Impact
-Attaque DOS	-Rendre le service indisponible. -Empêcher les utilisateurs légitimes d'utiliser un service.
-Brouillage du nœud dans le réseau de capteurs sans fil	-Blocage de la communication entre les nœuds.
-Ingénierie sociale	-Vol de données. -Propagation des malwares.
-Attaque par rejeu	-Violation de l'intégrité du système
-Attaque par canal latéral	-Dérober des informations sensibles.

TAB. 3.2 : Les attaques de la couche physique.

2. **La couche réseau :** L'objectif principal de la couche réseau de l'IdO est de transmettre les données collectées, les problèmes de sécurité dans cette couche sont liés à la disponibilité des ressources du réseau. Dans l'attaque de réseau, l'adversaire doit se concentrer sur le réseau du système IoT et l'attaquant n'a pas besoin être proche du réseau de l'IoT. Les mécanismes d'accès à distance et l'échange de données sensibles sur le canal sans fil augmentent la probabilité des attaques. Pour but de divulguer des informations privées et mettre en évidence des activités criminelles. -Le tableau 3.3 analyse brièvement les attaques de la couche réseau.

Nom de l'attaque	Impact
-Attaques d'analyse de trafic	-L'attaquant intercepte et examine les messages. -Divulgation de l'identité des communicants et le contenu de la communication.
-Man in the middle attacks	-Interception de la communication. -Obtenir des informations sensibles par l'écoute clandestine .
-Sinkhole attack (attaque de gouffre)	-Fuite de données des nœuds. -Suppression des paquets.
-RFID cloning	-Accéder aux données utiles par imitation RFID.
-Routing information attack	-Destruction du réseau par routage.
-Eavesdropping	-Perte de confidentialité des données.
-Sybil attack	-Donne à l'attaquant une autorité plus centralisée sur une plateforme décentralisée.
-DOS attack	-Rendre le service indisponible. -Empêcher les utilisateurs légitimes d'utiliser un service.
-RFID spoofing	-L'attaquant obtient un accès complet au système se faisant passer pour la source d'origine.

Tab. 3.3 : Les attaques de la couche réseau.

3. **Attaques de la couche application (Software Layer Attacks) :** Cette couche définit toutes les applications qui utilisent la technologie IOT parmi les fonctions les plus importantes de cette couche sont le traitement des données provenant des différents utilisateurs et la fourniture des services en temps réel. Il existe de nombreux problèmes dans la couche application dans laquelle la sécurité est le problème.

-Le tableau 3.4 analyse brièvement les attaques de la couche application.

Nom de l'attaque	Impact
-Une attaque par phishing	-Obtenir les informations privées telles que le nom d'utilisateur, les mots de passe.
-Attaque DOS	-Bloquer les utilisateurs de la couche application en refusant les services.
-Software vulnerabilities	-Le vol de données

Tab. 3.4 : Les attaques de la couche application.

3.1.2 l'attaque de routage dans la couche réseau

Les attaques de routage dans la couche réseau sont une préoccupation majeure en matière de sécurité, car elles peuvent compromettre l'intégrité et la disponibilité des communications, entraînant des conséquences potentiellement graves pour les réseaux et les systèmes connectés.

Les attaques contre RPL

Le protocole de routage RPL, conçu pour les réseaux LLN, reste vulnérable à diverses attaques de sécurité malgré ses caractéristiques spécifiques. Cette section fournit un aperçu des attaques ciblant RPL et leur classification.

Linus Walgren et al. [54] ont examiné le protocole de routage RPL et ont constaté que garantir la sécurité des dispositifs 6LoWPAN connectés à IPv6/RPL est un défi majeur. Les auteurs de l'article ont effectué une analyse exhaustive des technologies IoT et de leurs nouvelles fonctionnalités de sécurité, qui peuvent être exploitées par les attaquants et les systèmes de détection d'intrusion (IDS). Une contribution majeure de cet article est la démonstration d'une attaque de routage connue visant les réseaux 6LoWPAN utilisant le protocole de routage RPL. Ils ont mis en œuvre ces attaques sur l'implémentation RPL du système d'exploitation Contiki, puis les ont démontrées sur le simulateur Cooja. De plus, l'article met en évidence les nouvelles fonctionnalités de sécurité du protocole IPv6 et explique comment ces fonctionnalités peuvent être utilisées pour détecter les intrusions dans les dispositifs IoT en implémentant un protocole de pulsation léger.

Anass Rghioui et al. [44] Ce document présente une revue des problèmes de déni de service dans le réseau 6LoWPAN, en mettant l'accent sur les attaques ciblant son protocole de routage sous-jacent RPL. Le document présente en détail le fonctionnement de RPL, en particulier la façon dont il construit son schéma de routage en construisant un DoDAG. Il examine les problèmes de sécurité de RPL résultant des faibles ressources de 6LoWPAN, en se concentrant sur les attaques de déni de service (DoS) car elles sont les plus nuisibles. les menaces présentées dans ce document sont les attaques : identity, sybil, selective forwarding, blackhole, wormhole, sinkhole, hello flood and overload attacks.

Bouaoudia et al. [15] ont étudié Le Rank Attack (RA) de RPL qui est une attaque qui vise la propriété de rang utilisée pour sélectionner les parents dans la topologie RPL, et qui se produit au niveau de la couche réseau. En modifiant la valeur de rang, un nœud malveillant peut compromettre les performances de l'ensemble du réseau IoT. Il existe deux types d'attaques À : la première consiste à diminuer le rang RPL afin que le nœud malveillant soit choisi comme parent préféré par ses voisins. Ensuite, il peut exécuter diverses attaques telles que le Blackhole ou la modification des données.

Le deuxième type de l'attaque rank RPL est l'augmentation du rang RPL pour forcer ses voisins à chercher un autre parent ce qui va les pousser à augmenter leurs consommations énergétiques. De plus, c'est toute la topologie du réseau qui sera modifiée qui est étudiée dans [14].

Boudouaia et al. [14] dans leur article ont étudié les attaques de la couche réseau plus précisément les attaques contre RPL, ils les ont classées en trois catégories en fonction de leur impact sur la topologie. La première catégorie vise le trafic de données tel que les attaques de sniffing et d'usurpation d'identité. La deuxième vise les ressources de la topologie (par exemple l'énergie et la mémoire) afin de réduire la durée de vie des capteurs, qui sont déjà limités en termes de ressources. La dernière catégorie vise la topologie du réseau en insérant des nœuds malveillants qui cherchent à sous-optimaliser le réseau par un comportement malveillant. Les différents types d'attaque de rang en RPL sont présentés comme suit :

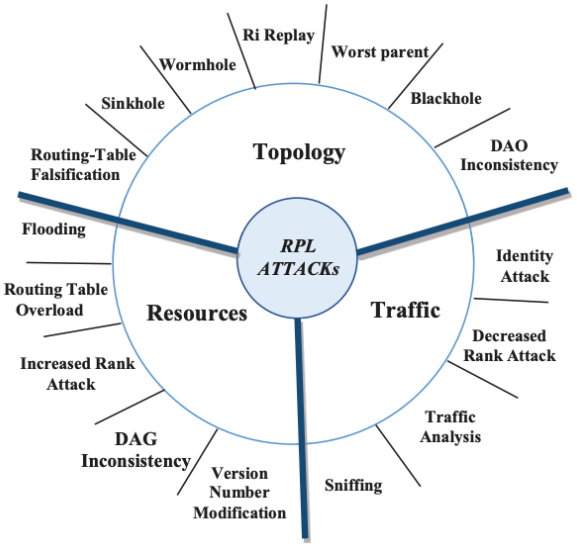

Fig. 3.2 : Classification des attaques RPL [14].

RPL, comme tout autre protocole de réseau de capteurs sans fil, s'est avéré vulnérable aux attaques de routage pour cela David Airehrour et al. [9] proposent une approche de sécurité afin de protéger le protocole de routage RPL contre ces attaques et précisément les attaques de "trou noir". Cette attaque est définie comme une attaque dans lesquelles un nœud malveillant se présente comme le routeur le plus approprié pour une destination donnée, puis ne parvient pas à transférer les paquets entrants, les faisant disparaître dans un trou noir. Dans ce cadre, les auteurs proposent un mécanisme qui utilise un algorithme de sélection de route qui prend en compte le niveau de confiance du nœud lors du choix de la meilleure route. La préférence est donnée aux nœuds avec un haut niveau de confiance pour le routage des paquets. Les résultats de l'évaluation expérimentale du mécanisme proposé montrent qu'il est efficace pour détecter et prévenir les attaques de trous noirs dans les réseaux RPL. Ce mécanisme permet de garantir un niveau de qualité de service des communications dans le réseau même en présence de nœuds malveillants.

Yavuz, F et al. [56] proposent une méthode de détection des attaques de routage pour l'IdO basée sur l'apprentissage approfondi. Dans leur étude, ils utilisent le simulateur Cooja-IdO, afin de générer des données d'attaque hautement précises dans des réseaux IoT dont la taille allant de 10 à 1000 nœuds. Ils proposent une méthode de détection

d'attaque basée sur l'apprentissage en profondeur hautement évolutive qui détecte les attaques de routage IoT telles que les attaques de catégorie restreinte, de type "hello-flood" et de modification de numéro de version, avec une précision élevée. Dans ce cadre, ils ont développé un réseau neuronal profond du modèle formé à l'aide de l'ensemble de données IRAD, qui contient des informations d'évaluation telles que la précision, l'exactitude et le taux de recherche. Ils ont finalement pu atteindre jusqu'à 99% sur la base des résultats F1 et des résultats des tests AUC.

Anhtuan Le et al. [34] ont évalué les performances du protocole RPL en analysant l'impact de quatre attaques potentielles que des attaquants internes pourraient utiliser : les attaques de type "rank", "Local Repair", "Neighbour" et "DIS". Les simulations ont révélé que ces menaces internes peuvent avoir des conséquences significatives sur le réseau RPL, telles que la réduction du taux de livraison, l'augmentation du délai de bout en bout et une surcharge accrue du contrôle, consommant ainsi davantage de ressources réseau. Les attaques "Decreased Rank" et "Local Repair" ont particulièrement affecté le taux de livraison, tandis que l'attaque "DIS" a entraîné une augmentation notable de la latence E2E. En revanche, l'attaque "Neighbor" a eu un impact minimal sur le réseau. Il convient de noter que l'étude a exclu l'effet de ces attaques sur la consommation d'énergie, car elle se concentrait uniquement sur le mode non sécurisé de RPL. Les résultats de cette étude suggèrent que les anomalies de performance peuvent révéler la présence d'attaques en cours sur le réseau. Les auteurs ont proposé d'utiliser ces résultats pour développer un système de détection d'intrusion basé sur l'anomalie de performance de RPL. Leur prochaine étape consistera à utiliser ces résultats comme ensemble d'entraînement pour développer un module IDS capable de détecter différents types d'attaques internes. Ainsi, cette recherche ouvre la voie à de futures avancées dans la sécurisation des réseaux RPL contre les attaques internes.

Dans [40], Anthea et al. ont étudié et examiné les attaques contre le protocole RPL IoT, en classant les attaques de routage possibles contre ce protocole. Ils ont classé trois catégories principales d'attaques, contre les ressources, la topologie et le trafic de données, la gestion des risques de ces attaques a été abordée. Les auteurs de cette étude n'ont pas réalisé de tests et de simulations d'attaques dans un environnement réel, ce qui aurait permis d'évaluer de manière précise l'impact néfaste de ces attaques sur les réseaux IoT.

Abhishek Verma et al. [53] ont analysé quelques attaques de routage bien connus telles que (Sinkhole, Blackhole, Selective forwarding, Sybil, Clone ID, HELLO flooding et Local Repair) et ont montré leur effet sur le réseau 6LoWPAN. Les résultats de la simulation permettent de conclure que les attaques de routage perturbent fortement le débit du réseau. Par conséquent, le nombre croissant d'attaques sur l'internet des objets peut perturber l'ensemble du réseau d'appareils intelligents.

Kumar et al.[33] ont simulé l'attaque Blackhole sur un réseau basé sur RPL afin d'en étudier les effets. Comme cela était prévisible, l'attaque a eu pour conséquence la diminution du PDR, ainsi qu'une augmentation à la fois de la latence E2E et de l'overhead des messages de contrôle. Cependant, les auteurs n'ont pas pris en compte la consommation d'énergie et n'ont pas considéré l'existence des mécanismes de sécurité de RPL.

Après toutes les recherches et enquêtes sur les attaques RPL, certains auteurs n'ont simulé aucune attaque de routage, tandis que d'autres n'ont simulé qu'une ou deux attaques. Par conséquent, sur la base de travaux antérieurs, A Krari et al [32]. ont proposée une nouvelle étude pour examiner et analyser les performances de RPL sous différentes attaques. La nouvelle recherche effectuera cinq attaques différentes sur le protocole RPL, à la fois directes et indirectes, en se concentrant sur des indicateurs de compromis tels que la perte de paquets, la consommation d'énergie, l'écoute clandestine.

Les auteurs de [52] ont présenté une solution appelée Secure-RPL pour contrer les attaques de type "DIS flooding" contre les réseaux 6LoWPAN basés sur RPL. Cette approche consiste à prévenir les nœuds légitimes de réaliser des réinitialisations inutiles du "trickle timer" et des transmissions DIO afin de préserver les ressources des nœuds. Pour détecter les intrusions, plusieurs données sont collectées, telles que l'adresse IP de l'émetteur, l'heure de réception du dernier message DIS et le nombre total de messages DIS reçus depuis la dernière réinitialisation. L'idée principale derrière Secure-RPL est d'utiliser les paramètres RPL pour établir des seuils de sécurité. Ces seuils de sécurité restreignent les réinitialisations inutiles du "trickle timer" et réduisent les transmissions de messages de contrôle résultant des attaques de type "DIS flooding". Les auteurs ont testé leur solution sur des réseaux de petite taille comprenant 8 et 16 nœuds, avec un seul attaquant.

Mehdi Rouissat et al.[45] ils ont proposé et étudié une attaque par modification du numéro de version (VNA), dans laquelle le nœud attaquant inonde le réseau avec des numéros de version falsifiés et incrémentés de manière continue. Ils ont présenté l'impact de cette attaque. Les résultats montrent que cette attaque modifiée entraîne une augmentation significative des frais généraux et de la consommation d'énergie, ainsi qu'une dégradation du taux de livraison des paquets et de la latence. Ils ont simulé l'attaque sous contiki 3.0 en utilisant les noeuds Z1, sous differents scenarios et conditions.

3.2 Taxonomie des attaques sur RPL

La taxonomie des attaques de routage dans les réseaux IoT est présentée dans la figure 3.3 et se divise en trois catégories principales. Dans cet article, les attaques de routage dans les réseaux IoT ont été largement classées en trois catégories [50] :

1. **Attaques contre les ressources du réseau :** celles-ci visent à faire consommer à un nœud légitime ses ressources énergitiques, de traitement ou de mémoire dans le but de perturber la disponibilité du réseau. Ces attaques sont particulièrement dangereuses pour les réseaux contraints, car elles réduisent considérablement la durée de vie des appareils et donc du réseau RPL. On peut distinguer deux grandes catégories d'attaques contre des ressources :

    - **Attaques directes :** où le nœud malveillant provoque directement une surcharge pour perturber le réseau. Par exemple : attaques d'inondation (flooding) et attaques de surcharge de la table de routage (routing table overload).

- **Attaques indirectes :** où le nœud malveillant incite les autres nœuds à générer de la surcharge. Comme les Attaques d'augmentation du rang (increasing rang attack), Attaques d'incohérence de DAG (DAG inconsistency) et les attaques par modification du numéro de version (version number modification).

2. **Attaques contre la topologie du réseau :** ces attaques ont pour objectif de perturber la structure du réseau RPL. Les attaquants cherchent soit à optimiser de manière sub-optimale la topologie du réseau, soit à isoler un groupe de nœuds RPL du reste du réseau. Cette catégorie peut également être classée en deux sous-catégories différentes en fonction des conséquences qui en résultent :

   - **La sous-optimisation :** qui signifie que le réseau convergera vers une forme non-optimale, induisant de mauvaises performances, comme : attaque de falsification de table de routage (routing table falsification), Attaque de puit (sinkhole), wormhole et (Routing Information Replay Attacks).

   - **L'isolation :** d'un nœud ou un ensemble de nœuds, les coupant du reste de la topologie RPL, y compris du nœud racine. Comme : l'attaque de trou noir (Blackhole) et Les Attaques d'incohérence DAO (DAO Inconsistency Attacks).

3. **Attaques contre le trafic réseau :** cette catégorie concerne les attaques contre le trafic réseau qui vise généralement à capturer les informations transmises par les noeuds. Telles que les attaques de spoofing ou les attaques de tromperie. Cette catégorie se subdivise à nouveau en deux sous-catégories en fonction de l'objectif poursuivi :

   - **L'écoute (Eavesdropping Attacks) :** des informations qui sont transmises par le réseau pour recueillir le trafic du réseau comme : sniffing et l'analyse du trafic du réseau.

   - **Le détournement :** d'un nœud ou d'un ensemble de nœuds, notamment pour altérer les informations légitimes échangées, comme les attaques du rang diminué (Decreased Rank Attacks) et les Attaques d'identité (Identity Attacks) .

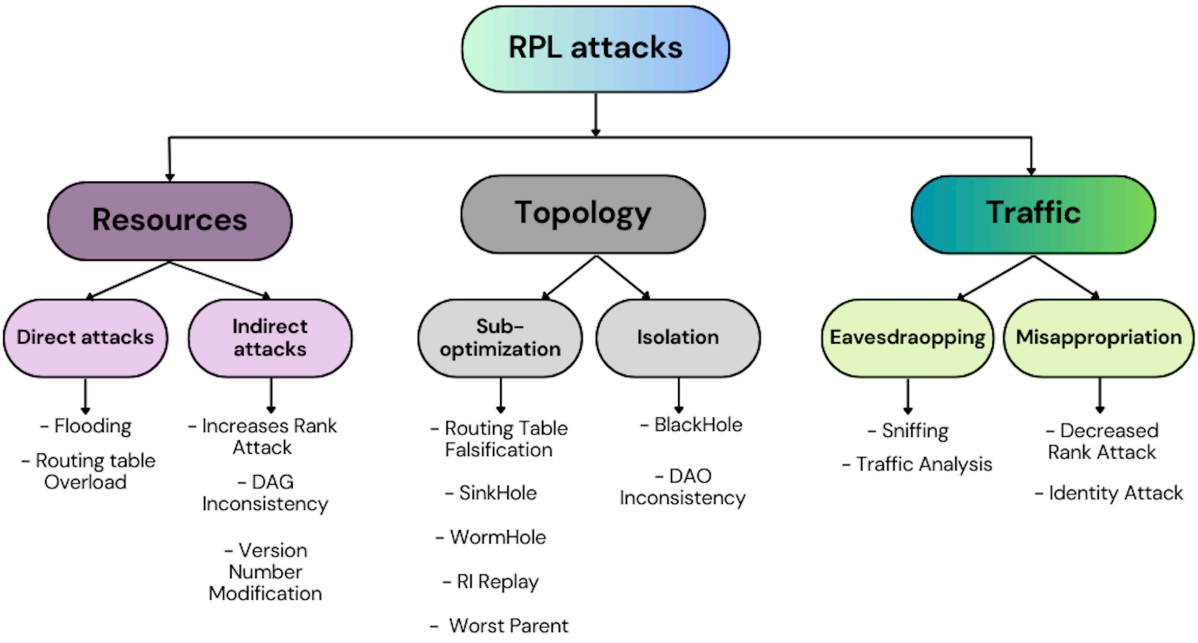

Fig. 3.3 : Taxonomie des attaques sur le RPL [50].

Au cours de la recherche bibliographique de ce mémoire, nous avons trouvé plusieurs études sur les vulnérabilités de RPL, leurs impacts et des solutions proposées pour empêcher quelques attaques, nous les résumons dans le tableau suivant.

Travail	Attaque	Contre-mesure	Evaluation
Anass Rghioui et al [44].	Analyse des attaques DOS et ses menaces tells que : Sinkhole, spoofing, sybil, selective forwarding, wormhole,hello flood.	Surveiller la force du signal, taux d'envoi de paquets, packet delivery ratio, packet send ratio, packet dropping rate et packet forwarding rate.	Implémentation d' un IDS pour faire face aux attaques DOS.
Bouaoudia et al [14], [15].	Rank attack (diminution et augmentation du rang et l'attaque du pire parent).	/	Une comparaison entre plusieurs attaques a été optée en utilisant le test Friedman
Airrrehour [9], Kumar et al [33].	Blackhole	Proposition d'un protocole de routage basé sur la confiance (algorithme de sélection de route) [9]	L'attaque a eu pour conséquence une diminution du PDR et une augmentation de la latence E2E et de la surcharge des messages de contrôle [33].
Linus Wallgren et al [54].	Implementation de :Wormhole, Hello Flood, Sinkhole, Selective Forwarding, Sybil	Placement d'un IDS et implémentation d'un protocole léger (heartbeat protocol)	Evaluer la consommation d'énergie et d'électricité d'un nœud.
Yavuz et al [56].	Hello flood, decrease rank et version number modification.	Proposition d'une méthode de détection des attaques basée sur l'apprentissage approfondi	Ils ont obtenu des chiffres élevés de performances basés sur le F1-score et AUC test score.
Anhtuan et al [34].	Rank, Local repair, Neighbor et l'attaque DIS.	/	L'impact des attaques sur les performances du réseau : le délai de bout en bout, le taux de livraison et la surcharge de contrôle générée.
Krari et al [32].	Comparaison et implémentation des attaques DAO, version number modification, blackhole, DIO flood et DIS.	/	Prouver l'impact de ces attaques sur les mesures de performances de RPL : power consumption, radio consumption et lost packet.
Verma et al [52].	Attaque DIS	un schema d'atténuation des messageds DIS appelé Secure-RPL	IL a été observé que l'attaque DIS augmente la surcharge des paquets de controle et la consommation d'énergie.

Tab. 3.5 : résumé de l'état de l'art.

3.3 Conclusion

Nous avons exposé, en ce chapitre, une classification des attaques ciblant le système IoT selon l'infrastructure de l'IdO, ensuite, on a focalisé notre recherche sur les attaques de routage dont on a présenté une taxonomie des attaques sur le protocole RPL en classant ces attaques dans trois catégories principales.

Dans le chapitre suivant, nous allons mettre en œuvre quatre attaques : Blackhole, Sinkhole, DIS et numéro de version dans un environnement de simulation Cooja Contiki pour illustrer les vulnérabilités du protocole RPL dans les réseaux IoT. Cette approche pratique nous permettra de mieux comprendre les mécanismes et les conséquences des attaques.

Chapitre 4

Simulation

4.1 Introduction

Dans ce chapitre, nous nous intéresserons à diverses attaques courantes qui ciblent le protocole RPL notamment l'attaque blackhole, sinkhole, DIS et numéro de version. Nous allons décrire le processus de la réalisation de notre simulation en utilisant Cooja. L'objectif est de simuler des scénarios réalistes afin de mieux comprendre les mécanismes, les vulnérabilités et les conséquences de ces attaques sur les réseaux IoT. Nous allons utiliser la solution VMware Fusion pour exécuter une machine virtuelle Vmware Fusion, nommée Instant ContikiOS.

4.2 Les outils de simulation

4.2.1 VMware Fusion

Est un logiciel de virtualisation développé par VMware spécifiquement conçu pour les utilisateurs de Mac. Il permet aux utilisateurs d'exécuter des systèmes d'exploitation Windows, Linux et d'autres systèmes d'exploitation sur leur Mac sans avoir à redémarrer l'ordinateur.

Le logiciel VMware Fusion est facile à utiliser avec une interface conviviale qui permet aux utilisateurs de créer et de gérer des machines virtuelles en quelques clics. Il fournit également des fonctionnalités avancées telles que l'exécution de machines virtuelles en mode plein écran, le partage de fichiers et de dossiers entre les systèmes d'exploitation hôtes et invités et la prise d'instantanés pour revenir aux configurations précédentes. Fusion est suffisamment simple pour les utilisateurs à domicile et suffisamment puissant pour les professionnels de l'informatique, les développeurs et les entreprises [5].

Nous allons utiliser la solution VMware Fusion pour exécuter une machine virtuelle sur MacBook Pro nommé Instant ContikiOS.

4.2.2 Contiki OS et son architecture

Contiki est un système d'exploitation léger, portable, flexible et open-source développé pour les nœuds capteurs des réseaux de capteurs sans fil (WSN). Il a été écrit en langage C pour assurer une meilleure portabilité, ce qui lui confère une grande portabilité. Il a été développé par une équipe de chercheurs suédois en 2004. Contiki repose sur un noyau événementiel et offre la possibilité d'une multitâche préemptive au niveau des processus individuels. Une configuration typique de Contiki nécessite environ 2 kilo-octets de RAM et 40 kilo-octets de ROM, Pour économiser la mémoire, Contiki utilise un concept appelé Protothread, qui est une approche hybride entre le multi-threading et la programmation événementielle. Contiki prend en charge deux types de communication. Tout d'abord, il utilise une couche de communication appelée Rime, qui permet le dialogue avec les capteurs voisins ainsi que le routage. Rime offre une transmission fiable des données.

Ensuite, Contiki utilise une deuxième couche appelée uIP (micro-IP), qui est une version réduite de la pile TCP/IP [43].

L'installation complète de Contiki comprend de nombreuses fonctionnalités telles qu'un noyau multitâche, une multitâche préemptive, des protothreads, une pile TCP/IP, le support d'IPv6, une interface utilisateur graphique, un navigateur web, un serveur web personnel, un client Telnet basique, un économiseur d'écran et une virtualisation du réseau.

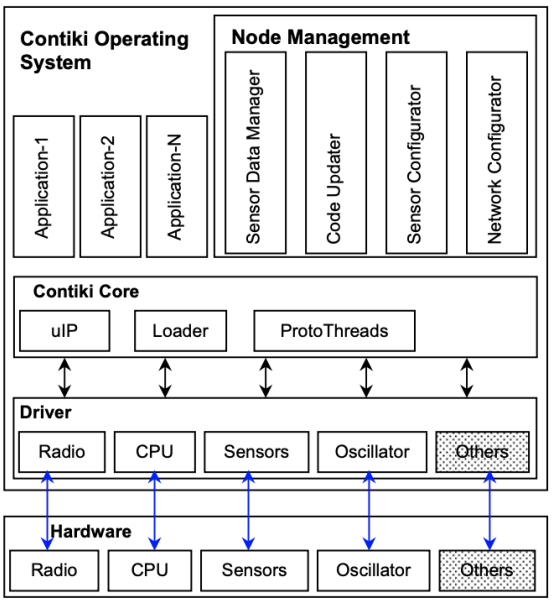

FIG. 4.1 : Architecture Contiki [22].

4.2.3 Le simulateur Cooja

Cooja est un outil de Contiki, c'est un simulateur de réseau qui permet l'émulation de plates-formes matérielles réelles. Vous pouvez simuler un réseau de capteurs sans fil en interagissant avec les nœuds du réseau. Cooja offre la possibilité de tester et d'évaluer des scénarios de réseau en émulant le comportement des nœuds et en permettant la communication entre les nœuds. Il permet aux développeurs et aux chercheurs d'analyser et de valider des protocoles, des algorithmes et des applications dans un environnement virtuel contrôlé avant de les déployer sur de véritables plates-formes matérielles [41].

L'interface de simulateur cooja est composée de plusieurs fenêtres (plugins) :

1. **Network :** offre la possibilité de visualiser les nœuds du réseau, y compris leur état (identifiant, adresse, LED, etc.). Au début de la simulation, cette zone est vide et nécessite l'ajout de nœuds pour pouvoir les afficher.

2. **Simulation Control :** la zone de contrôle est équipée des boutons Start (pour démarrer une simulation), Pause (pour arrêter la simulation), Step (pour régler la vitesse de la simulation) et Reload (pour recharger une simulation). Cette zone est utilisée pour contrôler la simulation, telle que le démarrage, le rechargement ou l'exécution pas à pas. Le temps d'exécution et la vitesse de la simulation y sont également affichés.

3. **Notes :** elle permet d'ajouter des notes à la simulation en cours, dans le but d'enregistrer des informations supplémentaires sur la simulation.

4. **Mote Output :** cette zone affiche les sorties des différentes interfaces des nœuds. On peut avoir une fenêtre "Mote Output" distincte pour chaque nœud.

5. **Timeline :** Il s'agit d'une chronologie qui affiche les actualités et les événements dans l'ordre chronologique. Il permet notamment de visualiser l'échange de données entre nœuds. Le message à afficher doit être spécifié par le code du nœud.

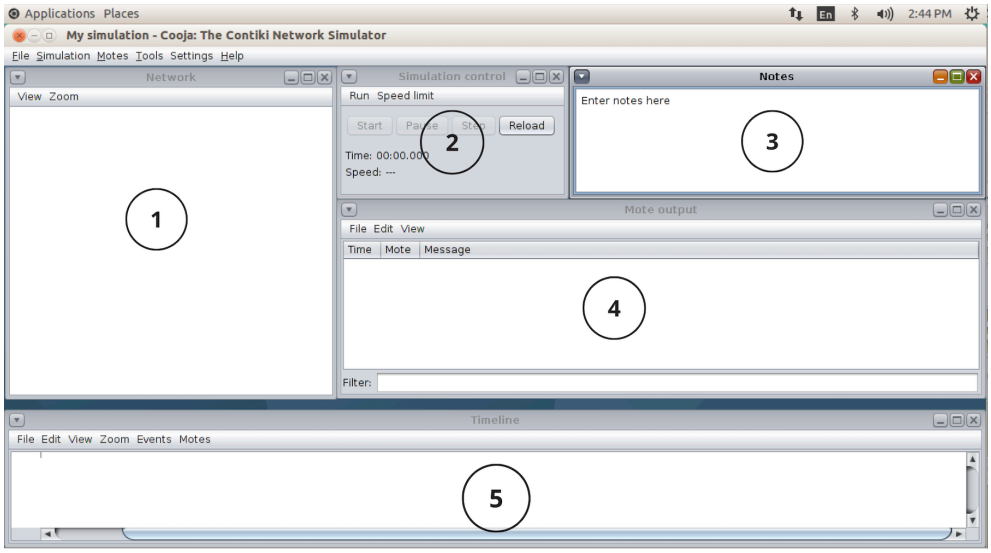

FIG. 4.2 : Les fenêtres cooja

4.3 Installation Instant Contiki

Instant Contiki est un environnement de dévéloppement simplifié, sous la forme d'une machine virtuelle Vmware, contient tout le code source de Contiki et toutes ses fonctionnalités ainsi le simulateur Cooja. Nous allons utiliser la solution VMware Fusion pour exécuter Contiki Instant sur MacBook Pro. Pour cela, suivez les étapes suivantes :

- Contiki Instant 3.0 peut être téléchargé à l'adresse suivante : https ://sourceforge.net/projects/cont
- Une fois le package zip est téléchargé, Il faut ensuite décompresser le fichier obtenu.
- Télécharger et installer VMware Fusion depuis l'url,
  https ://www.vmware.com/products/fusion/fusion-evaluation.html
- Ouvrir VMware Fusion et importer le fichier instant Contiki virtual machine(vmx), puis entrer le mot de passe par défaut : user
- Ouvrir le terminal de contiki et exécuter les commandes suivantes :
  - git submodule update -–init -–recursive
  - cd/contiki/tools/cooja

- enfin, pour obtenir le simmulateur cooja exécutez la commande suivante dans le terminale :

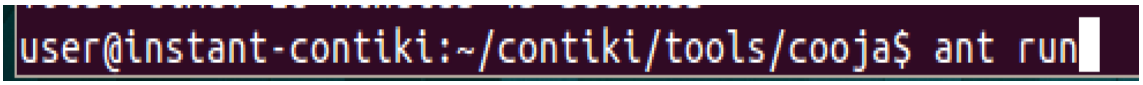

Fig. 4.3 : Commande d'exécution cooja

- La fenêtre cooja s'ouvre avec succès.

4.4 Paramètres et environnement de développement

4.4.1 Caractéristiques de la machine

La machine utilisée dans la simulation du réseau est caractérisée par les paramètres suivants :

Champs	**Valeur**
Processeur	2,3 GHz Intel Core i5 double cœur
RAM	8 Go
Disque Dur	SSD

Tab. 4.1 : caractéristiques de la machine utilisée.

4.4.2 Paramètres de simulation

Dans notre étude, nous avons utilisé l'exemple de rpl-collect et un environnement de simulation comme montre le tableau ci-dessous :

Paramètres	**Valeurs**
Simulateur	Cooja
Contiki	InstantContiki 3.0
Couche d'adaptation	6LOWPAN
Protocole de routage	RPL
Nombre de noeuds	15
Type de noeuds	Z1 Mote
Nbr de noeuds malveillant	1
Temps	10-15 min
Surface (mètres)	100 X 50

Tab. 4.2 : Paramètres de simulation.

4.4.3 Métriques de la simulation

Nous avons simulé 3 fois, le réseau initial pendant 15 minutes, puis nous avons mesurer ses performances pour observer l'impact de chaque attaque. Dans notre étude, nous avons choisi d'évaluer le protocole RPL en termes de :

- **Consommation moyenne d'énergie :** est une mesure de performance importante lorsque nous ciblons les LLN comme ses noeuds ont des contraintes de batterie.C'est la puissance moyenne consommée par tout les neouds du réseau (mW).Dans notre cas, elle est ontenue à l'aide de collect view.

- **Le nombre de messages de contrôle :** constitue le nombre total de messages DIO, DAO et DIS transmis dans le réseau pendant la simulation, pour suivre la surcharge du trafic. Un script Perl est utilisé pour extraire et calculer les messages.

4.5 Implémentation

L'objectif de notre expérimentation est de modéliser un réseau Low-Power and Lossy Network (LLN) en utilisant l'outil Cooja et la version 3.0 de Contiki et d'implémenter 4 attaques : attaque blackhole, Sinkhole DIS et version number afin d'étudier leurs impacts sur les performances du réseau, notamment en termes de la consommation d'énergie des noeuds, de surcharge du trafic et de la stabilité du réseau. Ce réseau sera composé de 14 nœuds clients et un noeud serveur.

Nous utilisons un plugin Cooja appelé Contiki Test Editor pour mesurer le temps de simulation et arrêter la simulation après le temps spécifié. Ce plugin crée également un fichier journal (COOJA.testlog) pour toutes les sorties du simulation que nous analyserons en fin de simulation à l'aide d'un script Perl. En évaluant les performances dans différents scénarios d'attaques, nous pourrons mieux comprendre les vulnérabilités du protocole RPL.

4.5.1 Réseau de référence

Nous avons mis en place un scénario de simulation d'un réseau LLN dans des conditions normales, en faisant varier 15 nouds. Tous les nœuds utilisent le code par défaut de Contiki et sont considérés comme légitimes. Ils transmettent leurs informations, telles que l'utilisation du CPU et la charge restante, vers le point de collecte, qui est le "sink" dans notre cas. Cette simulation nous permettra d'obtenir des données de référence sur le comportement normal du réseau, sans aucune attaque.

Objectif

Il est essentiel de mettre en place un réseau de référence en utilisant les paramètres spécifiés dans le tableau 4.2, afin d'obtenir une base de référence pour la comparaison.

Cette section se concentrera sur la méthodologie et les étapes nécessaires pour créer ce réseau de référence. En simulant un environnement de réseau LLN réaliste et en recueillant des données précises, nous pourrons évaluer l'effet des attaques sur la performance et l'efficacité énergétique du réseau. les résultant de ce scénario serviront de référence pour comparer et évaluer les résultats obtenus lors de la simulation d'une attaque.

Pour ce faire, nous avons utilisé le simulateur Contiki/Cooja avec une portée de transmission Tx de 50 mètres et une portée d'interférence INT de 100 mètres, en plus d'une distribution aléatoire des nœuds. Les détails de la configuration de la simulation normale, sans aucune attaque, ainsi que les emplacements des nœuds sont présentés dans la figure suivante.

Fig. 4.4 : La topologie de la simulation sans noeud malicieu.

Résultats et Discussion

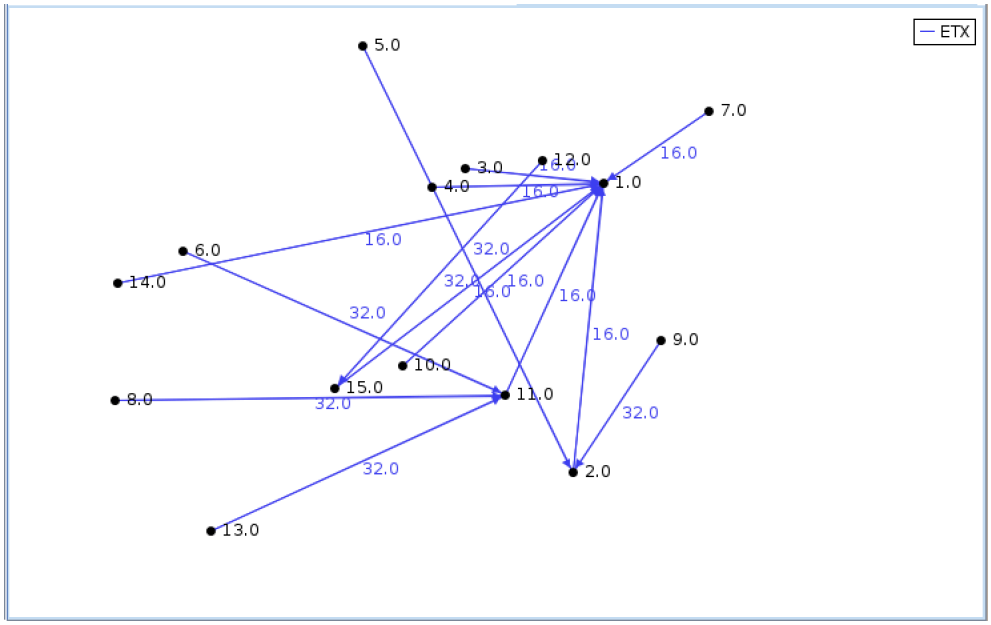

FIG. 4.5 : topologie graphique du réseau.

La figure ci-dessus représente une visualisation graphique des nœuds du réseau et de leur position géographique.

1. **Consommation d'énergie**

   Après avoir procédé à la simulation, à l'aide Collect View nous avons obtenu les résultats suivants :

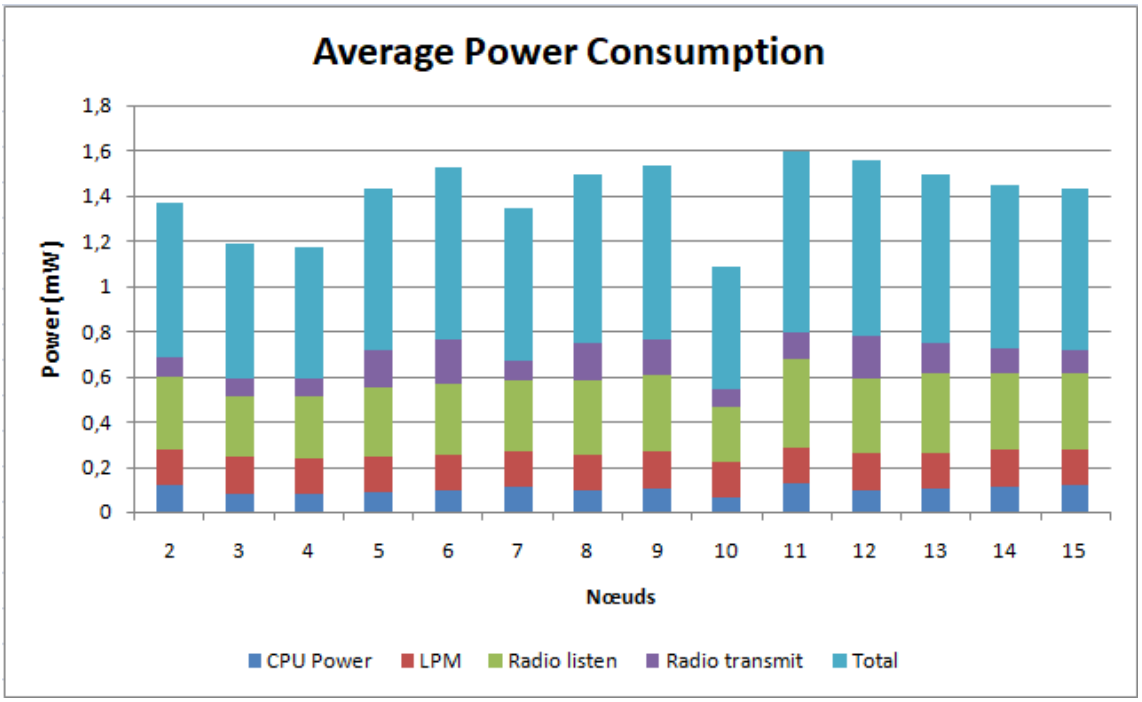

FIG. 4.6 : Le graphe de consommation d'énergie moyenne du réseau.

Cette figure présente les facteurs qui contribuent à la consommation d'énergie globale du nœud :

(1) **Radio Transmit (Transmission radio)** : indique la consommation d'énergie associée aux périodes où le nœud émet des transmissions radio (i.e. le nœud est actif).

(2) **Radio Listen (Écoute radio) :** représente la consommation d'énergie pendant les périodes où le nœud est en écoute de signaux provenant d'autres nœuds du réseau.

(3) **CPU :** représente la consommation d'énergie liée à l'activité du processeur du nœud.

(4) **LPM (Low Power Mode - Mode basse consommation) :** indique la consommation d'énergie lorsque le nœud est en mode basse consommation (mode inactif.

Comme on peut le constater dans les graphiques ci-dessus, à savoir la figure 4.6 aucun paramètre n'est affecté. La consommation moyenne d'énergie de tous les nœuds tourne autour de 0.796 mW. De plus, la consommation d'énergie liée à l'écoute radio et à la transmission radio est stable, les taux sont réguliers. L'écoute radio est logiquement plus élevée que la transmission radio, car les nœuds reçoivent divers messages de contrôle lors de la formation du DODAG.

2. **Messages de contrôle**

Message de contrôle	Sans attaque
DIS	14
DIO	258
DAO	99

Tab. 4.3 : Les message de contrôle dans le réseau sans attaque.

Tous les résultats que nous avons obtenus lors de la simulation standard sans attaques sont corrects et servent de référence pour le comportement attendu des nœuds. Par conséquent, ils peuvent être utilisés comme point de comparaison pour les résultats des autres simulations.

4.6 Implémentation des attaques

Cette section détaillera la mise en place de nœuds malveillants au sein du réseau de référence et expliquera comment déclencher des attaques. L'objectif de la simulation de l'attaque est de comprendre l'impact de la consommation d'énergie subi par les nœuds terminaux lorsqu'un nœud malveillant déclenche une attaque. Il sera également précisé que le nœud malveillant qui est dans notre cas le noeud 15 maintiendra une position précise pour chaque simulation afin de bien observer l'impact de chaque attaque.

4.6.1 Attaques sur la topologie

Attaque Blackhole

Dans ce cas précis, on a 15 nœuds dans la topologie, parmi ces nœuds, il y a un nœud ”sink” (récepteur) et 13 nœuds ”sender” (émetteurs). Il y a également un unique nœud ”blackhole” (ID = 15).
Le nœud attaquant prétend faussement avoir le chemin le plus court vers la destination en modifiant son rang de manière illégitime. Il refuse de transmettre les paquets de routage reçus de ses victimes et ne les propage pas vers le point de destination spécifié. Après avoir simulé cette attaque pendant une durée de 10 minutes (temps réel), nous analyserons et interpréterons les conséquences qui en découlent.

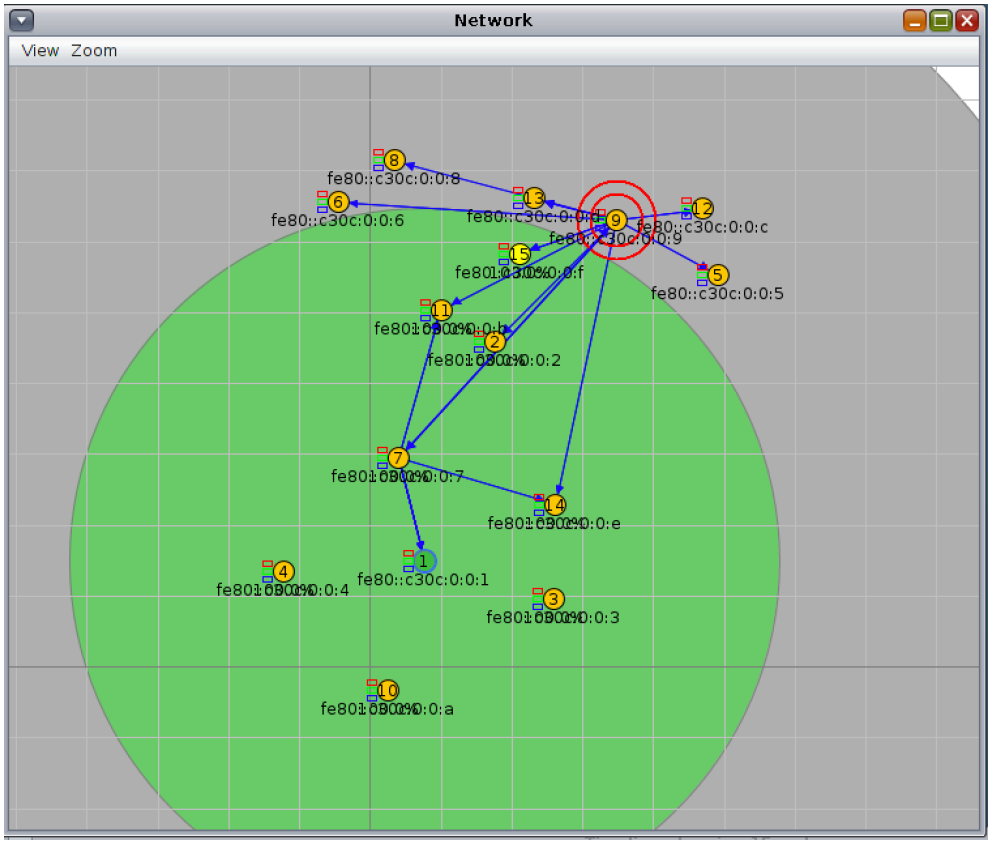

FIG. 4.7 : La topologie de l’attaque blackhole.

Résultats et Discussion

On remarque que l'attaque blackhole a supprimé tous les paquets que le nœud malveillant 15 est censé de transmettre, dont les nœuds 8, 13, 9, 12 et 5.
Lors de la simulation avec le nœud malveillant (15) qui effectue une attaque de type "blackhole" (trou noir). En observant la figure 4.8 on remarque l'absence de certains nœuds dans le graphe suivant, ces nœuds manquants ont été isolés par l'attaque blackhole, ce qui signifie qu'ils ne sont plus accessibles pour les autres nœuds du réseau.

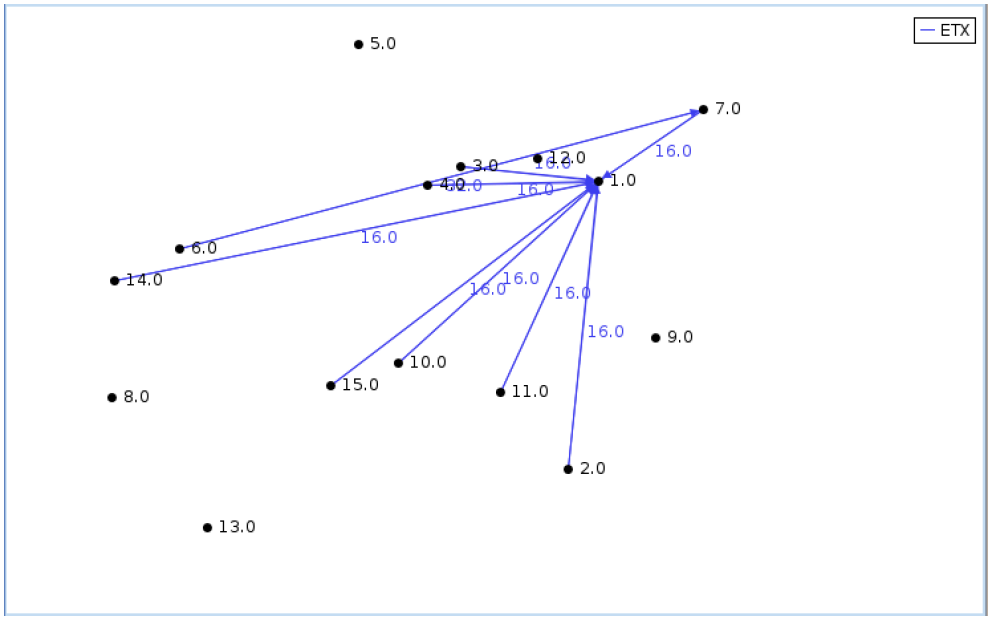

FIG. 4.8 : Topologie graphique du réseau avec l'attaque blackhole.

Les observations mettent en évidence les conséquences de l'attaque blackhole sur la structure du réseau. Certains nœuds sont isolés à la suite de cette attaque, ce qui les empêche de participer aux communications. Cette situation entraîne une diminution de la connectivité et perturbe le fonctionnement global du réseau.

1. **Consommation d'énergie**
   Ces résultats mettent en évidence l'impact spécifique de l'attaque blackhole sur les différents types de messages échangés dans le réseau, cela peut perturber le bon fonctionnement du réseau à cause de la suppression ou la perte de messages de contrôle importants.

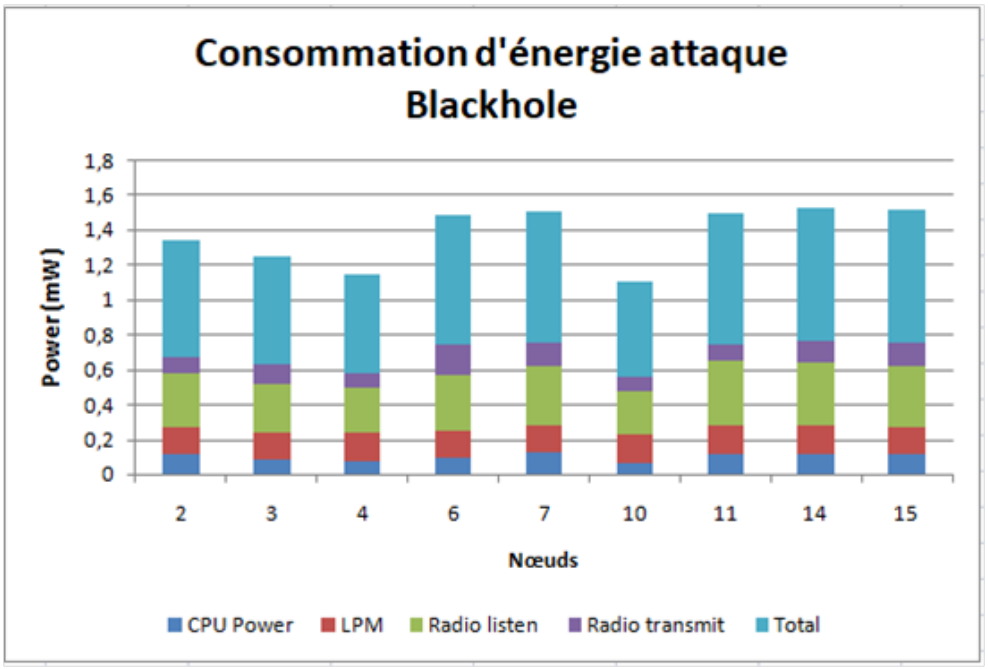

FIG. 4.9 : Le graphe de consommantion d'énergie moyenne de l'attaque blackhole.

Le graphique illustré dans la figure 4.10 présente les niveaux de consommation d'énergie pendant une attaque blackhole. Il met en évidence une augmentation de la consommation d'énergie pour certains nœuds, ce qui peut entraîner l'épuisement de l'énergie des nœuds environnants

2. **Messages de contrôle**

Le tableau suivant compare le nombre de messages DIO/DAO délivrés par chaque nœud dans deux scénarios : sans attaque et avec l'attaque activée.

Message de contrôle	Sans attaque	Avec attaque
DIS	14	14
DIO	258	255
DAO	99	93

TAB. 4.4 : Les message de contrôle avant et après l'attaque blackhole.

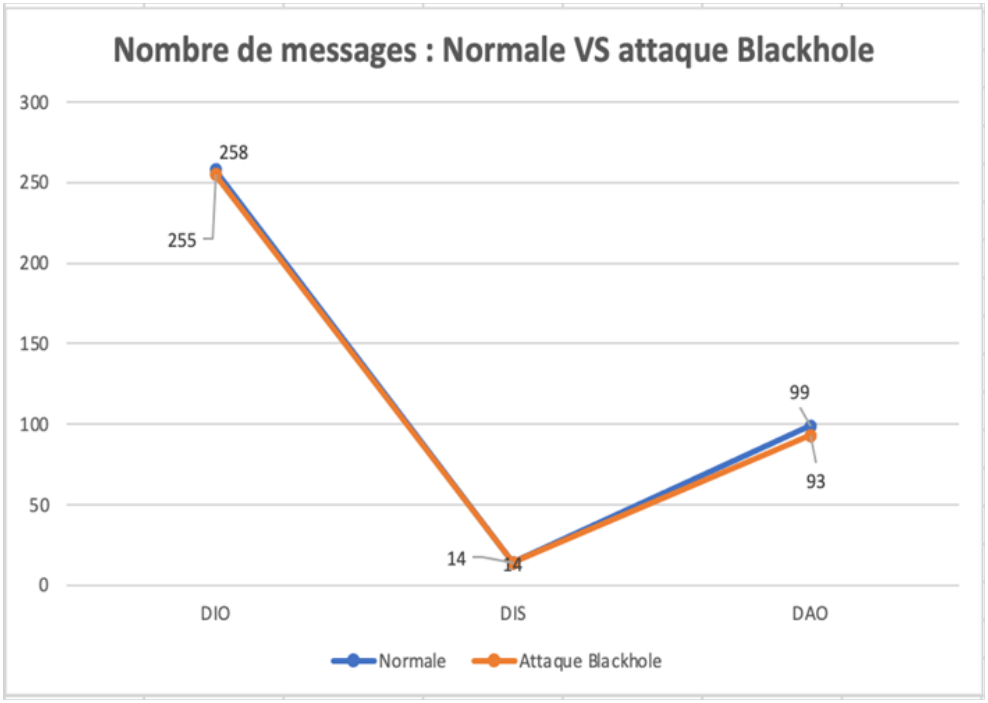

FIG. 4.10 : La courbe de comparaison des messages de controle.

Cette courbe démontre que l'attaque blackhole n'a aucun impact sur les messages DIS, tandis que le nombre de messages DIO et DAO diminue.

Attaque Sinkhole

Cette attaque se déroule en deux étapes distinctes. Tout d'abord, l'attaquant diminue intentionnellement son rang dans le réseau afin d'attirer les nœuds voisins. Ensuite, une exécution de l'attaque blackhole intervienne en supprimant tous les paquets que les voisins transportent.

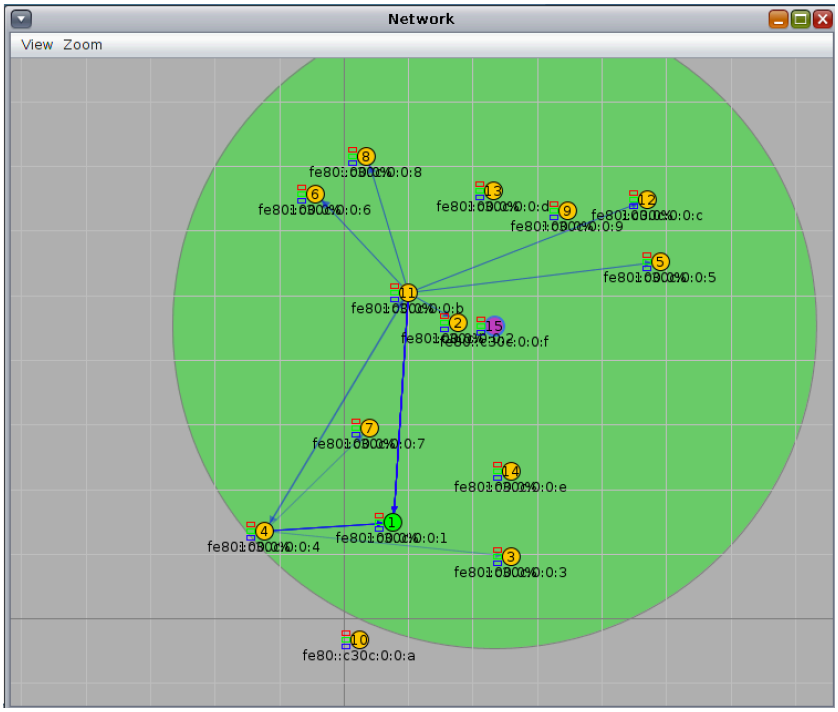

FIG. 4.11 : La topologie de l'attaque sinkhole.

Résultats et discussion

Après avoir simulé cette attaque sur une topologie de 15 nœuds (un nœud "sink", 13 nœuds "sender" et un unique nœud "sinkhole"), pendant une durée de 10 minutes(réel), nous analyserons et interpréterons les conséquences qui en découlent.

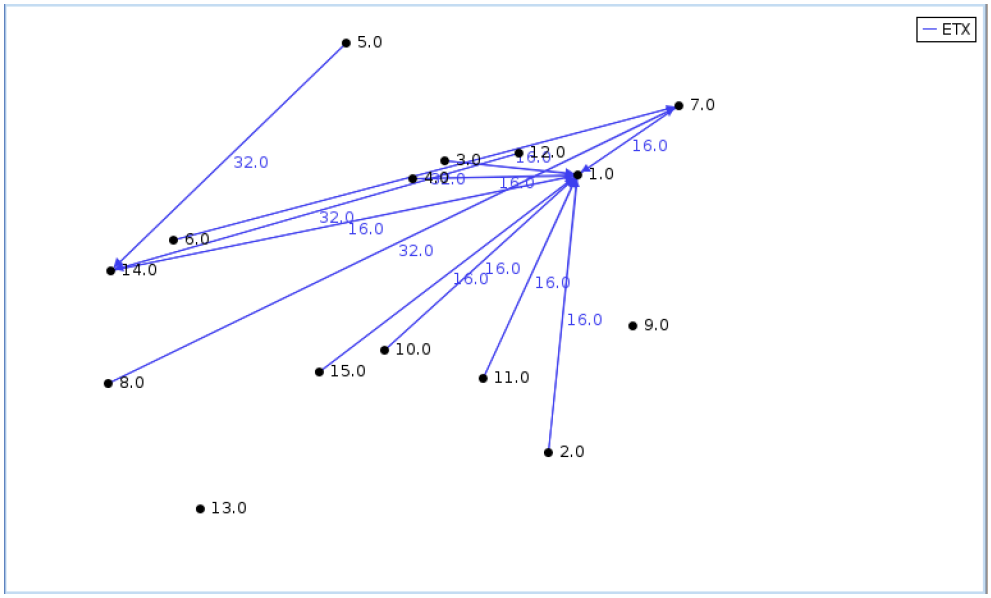

FIG. 4.12 : Topologie graphique du réseau avec l'attaque Sinkhole

En observant la figure 4.12 fournie, on remarque que l'attaque Sinkhole a engendré une perturbation majeure dans la topologie du réseau en attirant de manière sélective le trafic des nœuds voisins (nœuds 5, 6, 8, 9, 12 et 13) vers les Sinkholes. Cette sélection a

créé des routes non-fiables et instables, conduisant à la suppression des nœuds 9 et 13 qui ont choisis comme parents préférés. Toutefois, les nœuds 5 et 12 n'ont pas été supprimés car leur parent préféré était le nœud 14 et les noeuds 6 et 8 leur parent préféré était le nœud 7.

1. **Consommation d'énergie**

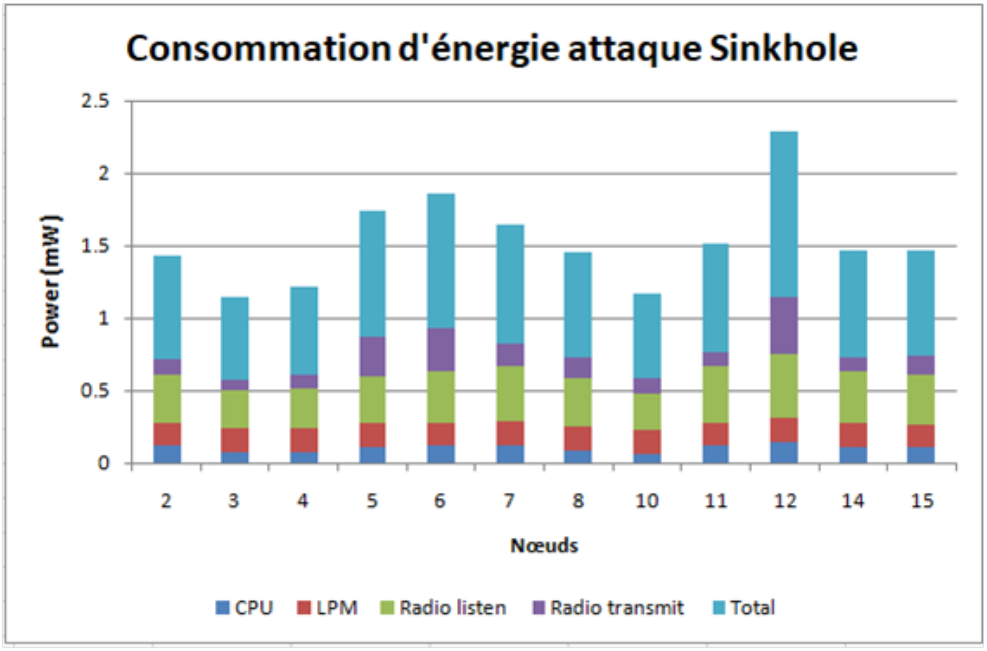

Fig. 4.13 : Le graphe de consommantion d'énergie moyenne de l'attaque Sinkhole

Comme nous pouvons le constater d'après la figure 4.13 La consommation d'énergie et de radio est également augmentée pour tous les nœuds, ce qui peut entraîner une épuisement de l'énergie des nœuds environnants.

2. **Messages de contrôle**

Ces résultats mettent en évidence que l'attaque sinkhole a un impact significatif sur la distribution des messages DIO et DAO dans le réseau. Ce qui signifie l'attaquant réussit à manipuler la structure du réseau en attirant les nœuds vers lui-même en tant que destination préférée, ce qui entraîne une augmentation du trafic de routage dans le réseau.

Dans le tableau ci-dessous, une comparaison est faite entre le nombre de messages DIO/DAO délivrés par chaque nœud dans deux scénarios distincts : un scénario sans attaque et un scénario avec l'attaque activée. Les résultats mettent en évidence que l'attaque sinkhole n'a aucun impact sur les messages DIS, tandis que le nombre de messages DIO et DAO augmente.

Message de contrôle	Sans attaque	Avec attaque
DIS	14	14
DIO	258	520
DAO	99	224

TAB. 4.5 : Les message de contrôle avant et après l'attaque sinkhole.

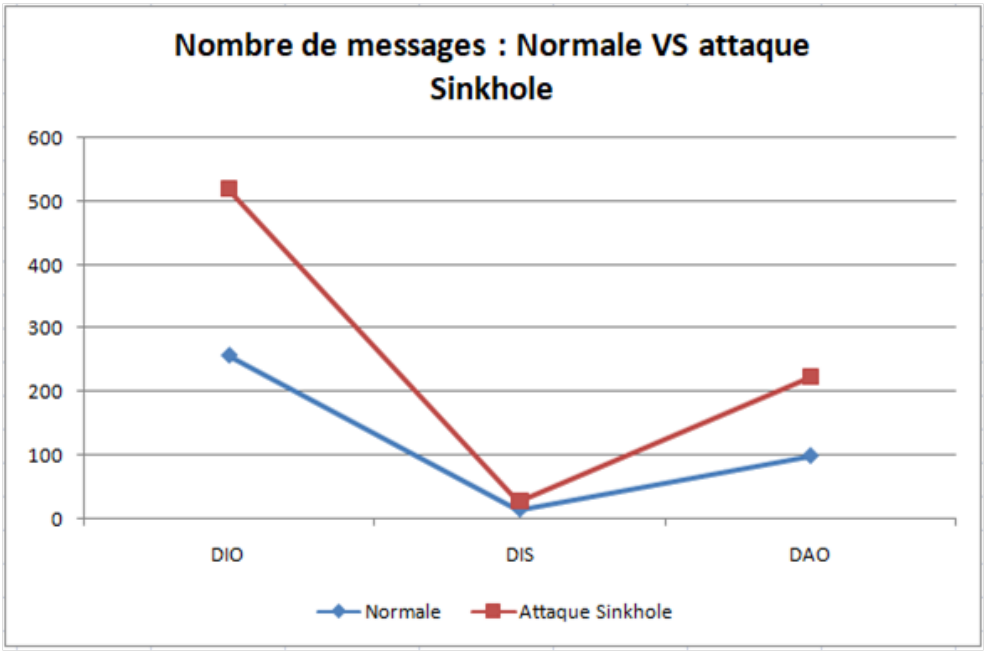

FIG. 4.14 : la courbe de comparaison de message de contrôle

4.6.2 Attaques sur les ressources

Attaque DIS

Elle consiste à générer une grande quantité du trafic dans le réseau dans le but de rendre les nœuds et les liens indisponibles. Pour simuler cette attaque, nous avons fixé l'emplacement du noeud malicieu 15 comme il est montré dans la figure 4.15.

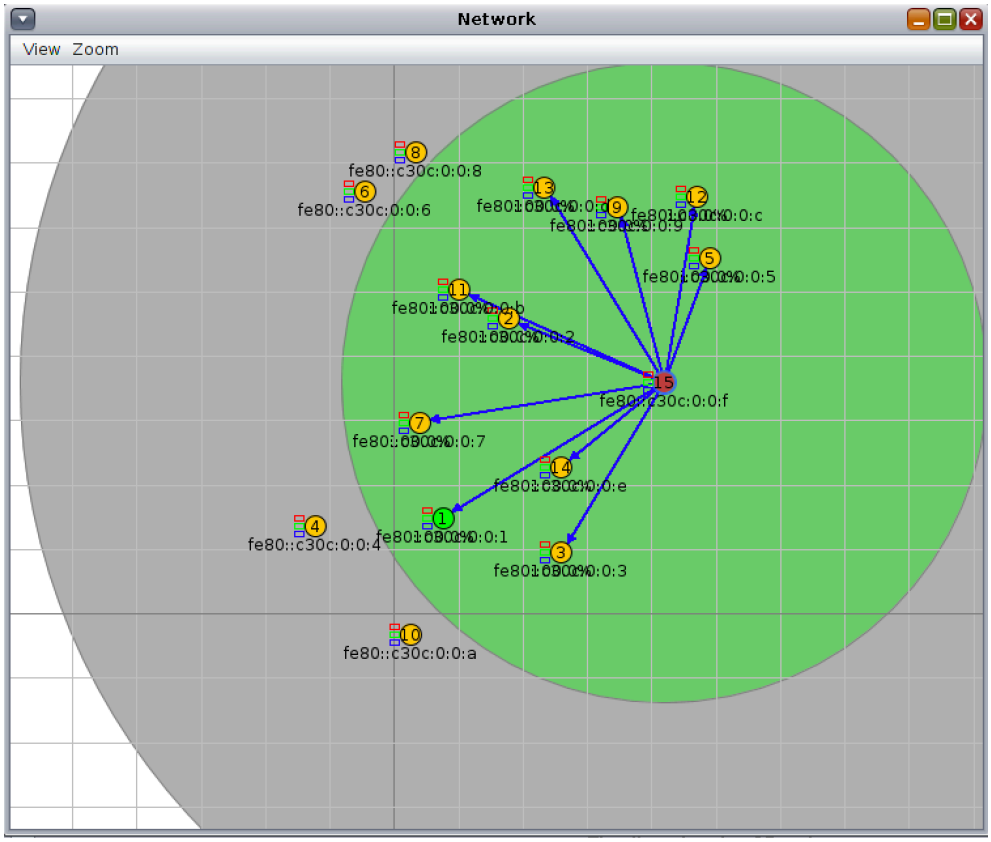

FIG. 4.15 : La topologie de simulation de l'attaque DIS.

Le noeud malicieu (15) enverra des messages DIS (DODAG Information Solicitation) à tous les nœuds du réseau, même ceux qui ne sont pas directement impliqués. En réponse à ces messages, les nœuds ciblés enverront des messages DIO (DODAG Information Object) au nœud 15. L'idée est que le nœud 15, étant nouveau dans le graphe DODAG, tente de rejoindre rapidement le réseau en envoyant des messages DIS à une liste spécifique de nœuds, y compris les nœuds : 5, 12, 9, 13, 11, 2, 7, 1, 14 et 3. Nous allons analyser et interpréter les conséquences de cette attaque dans ce qui suit.

Résultats et Discussion

1. **Consommation d'énergie**

   Le graphe suivant présente les résultats de la consommation d'énergie en fonction de la taille du réseau, après l'exécution de la simulation avec un nœud malveillant pendant 10 minutes (réelles).

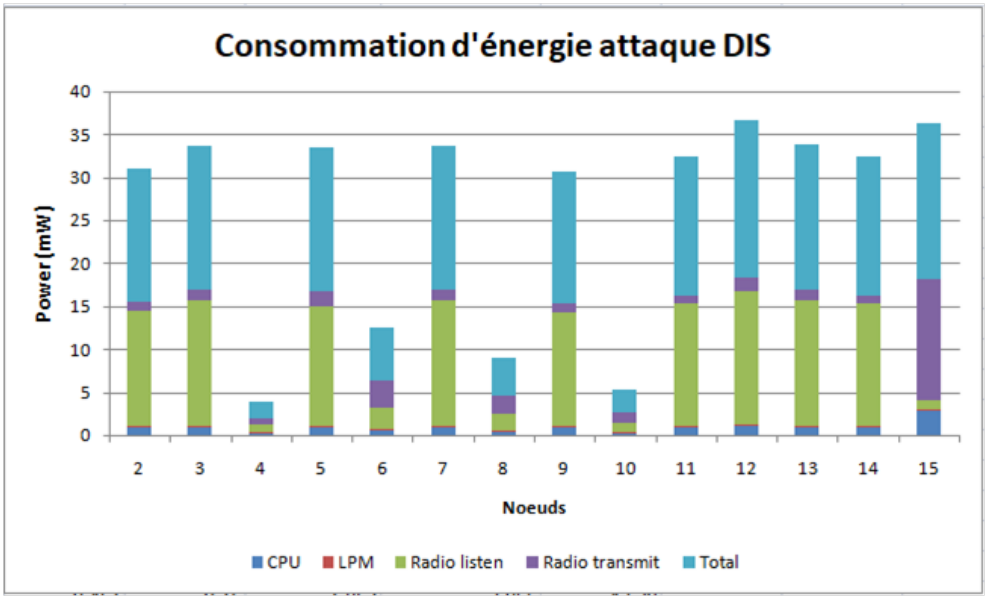

FIG. 4.16 : le graphe de la consommation d'énergie moyenne de l'attaque DIS

La répétition des messages envoyés a un impact direct sur la consommation d'énergie des nœuds du réseau. Cela est clairement illustré dans le graphique de la consommation d'énergie, où la consommation totale a atteint 37 mW. Ainsi, il est remarqué que les nœuds à proximité du nœud malveillant 15 ont les valeurs les plus basses pour leur mode de faible consommation d'énergie (LPM). Cette observation suggère que ce type d'attaque affecte directement les nœuds qui se trouvent dans la portée de transmission de l'attaquant.

Nous avons comparé les niveaux d'énergie entre la simulation normale et la simulation de l'attaque DIS. En revanche à travers la courbe de la figure ci-dessus qui représente la consommation d'énergie des deux cas nous constatons qu'il y'a une variation d'énergie entre les deux, On est passé d'une énergie de 0.796 au mW à 18.1 mW. L'attaque DIS Flooding agit énormément sur l'énergie, cela est due à la multiplication des transferts des messages de contrôle engendrés par le nœud malveillant.

2. **Messages de contrôle**

Le nombre total des messages de contrôle publiés par les noeuds dans le scénario est résumé dans le tableau suivant :

Message de contrôle	Sans attaque	Avec attaque
DIS	14	5976
DIO	258	708
DAO	99	428

TAB. 4.6 : le nombre de message de contrôle avant et après l'attaque DIS.

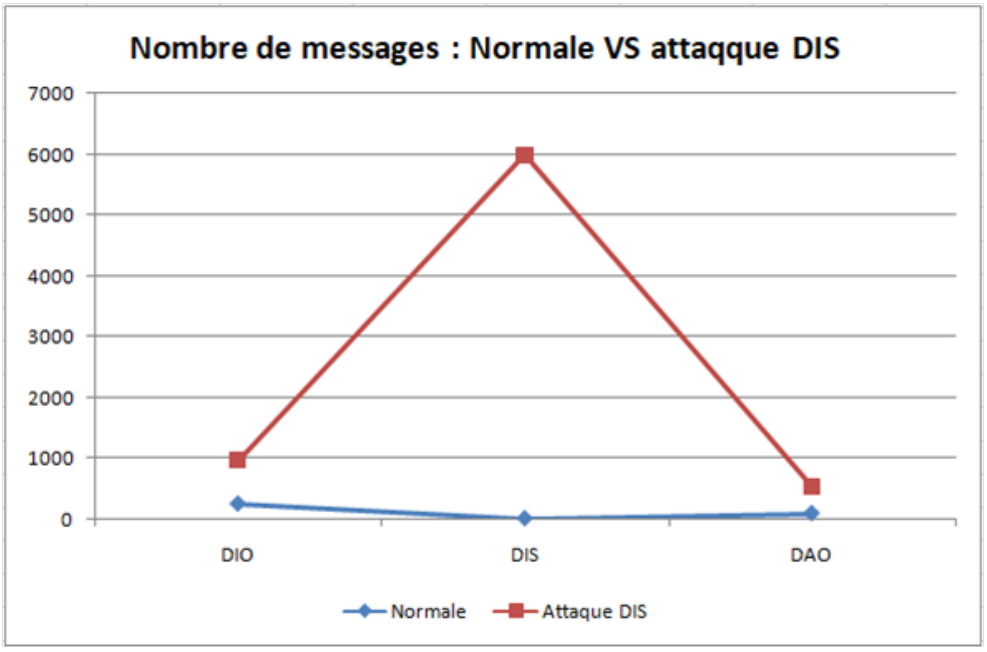

Fig. 4.17 : La courbe de comparaison des messages de contrôle.

La figure ci-dessus présente une courbe permettant de comparer le nombre de messages transmis dans un réseau sans attaque et dans un réseau où l'attaquant est présent. On observe un nombre considérablement élevé des messages DIS, passant de 14 à 5976 messages. Cela montre clairement l'impact de l'attaque DIS sur le réseau. Le nœud malveillant utilise les messages DIS pour solliciter des informations de tous les nœuds.En réponse à ces messages, les nœuds ciblés envoient des messages DIO au nœud 15, d'où l'augmentation des messages DIO de 258 à 708 DIO. C'est ainsi que le nœud 15 essaie de rejoindre rapidement le réseau.

Attaque Numéro de version

Dans cette attaque, nous avons considéré un attaquant unique (le noeud 15) qui augmente de manière inimitable le numéro de version du DODAG et le publie. Les nœuds destinataires supposent qu'une réparation globale est en cours et qu'une nouvelle version du DODAG est en train d'être créée , ils commencent donc à transmettre des messages DIO. Afin d'analyser l'effet de l'attaque du numéro de version, nous nous somme basés sur le scénario suivant :

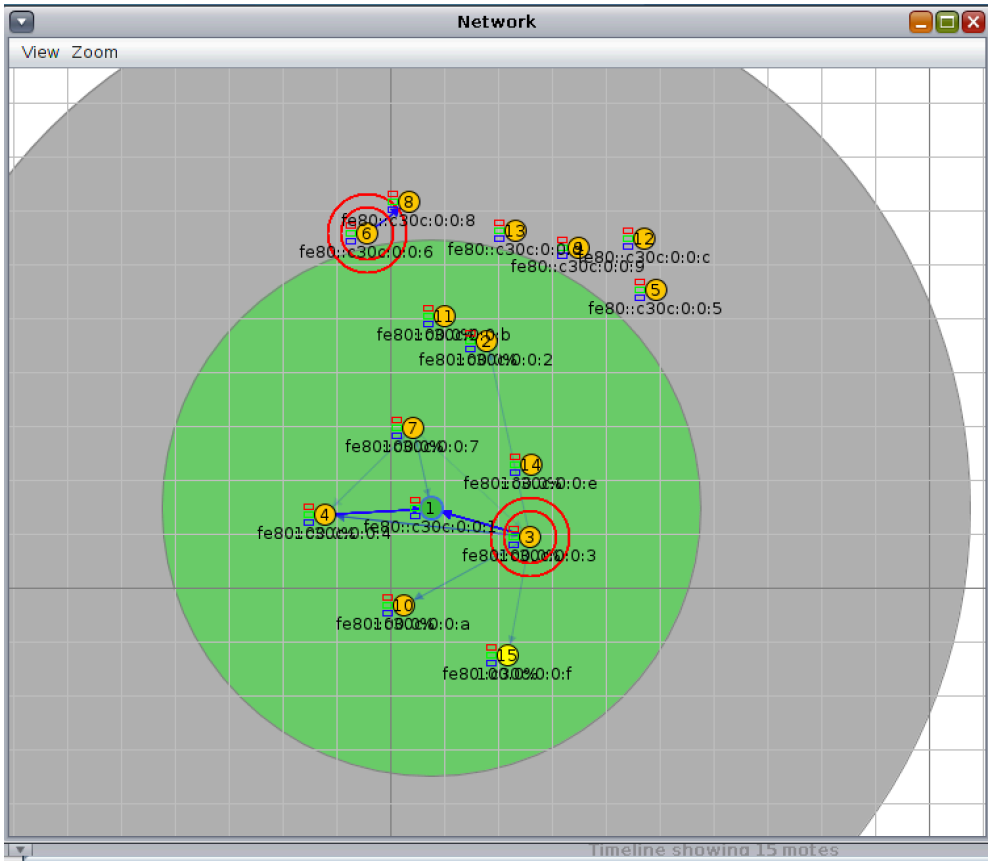

FIG. 4.18 : La topologie de simulation de l'attaque Version number.

Résultat et Discussion

1. **Consommation d'énergie** Dans cette section, nous évaluons l'impact de l'attaque numéro de version sur la consommation d'énergie totale consommée. En examinant les résultats de la simulation, nous constatons que l'attaque de modification du numéro de version a un impact significatif sur la consommation d'énergie liée aux opérations de transmission radio.

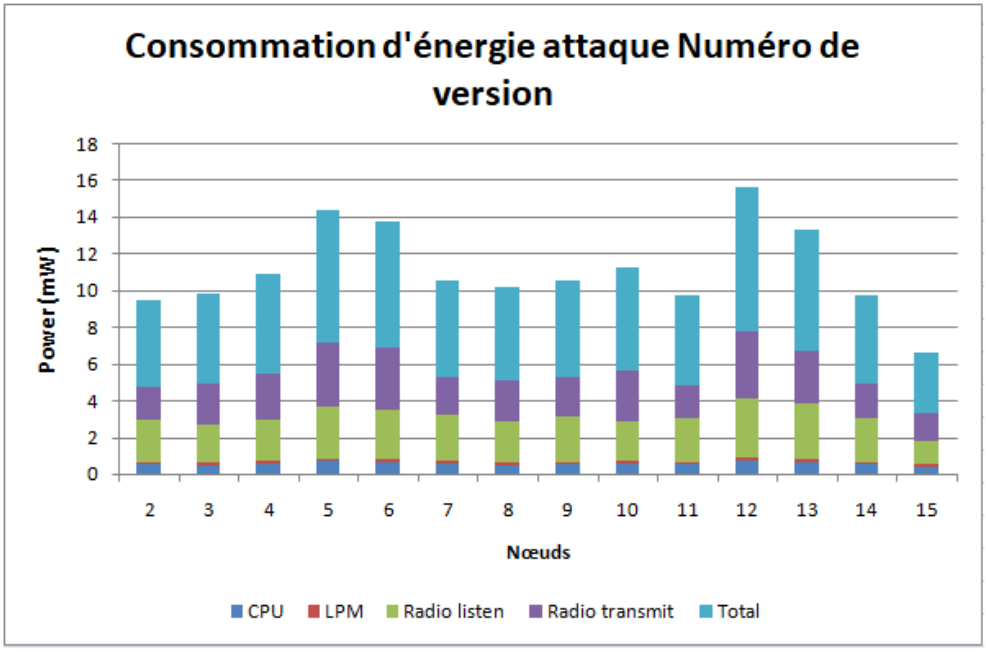

FIG. 4.19 : le graphe de consommation d'énergie moyenne de l'attaque Version number.

Lorsque le numéro de version est modifié, les nœuds du réseau doivent mettre en œuvre des mécanismes de rétablissement et effectuer des transmissions supplémentaires pour maintenir la connectivité. Cela entraîne une augmentation de la consommation d'énergie due aux opérations de transmission radio.

De même, l'écoute radio est également affectée par cette attaque. Les nœuds voisins qui détectent les changements du numéro de version doivent écouter plus fréquemment pour identifier les nouvelles routes disponibles et s'adapter aux modifications. Cela se traduit par une augmentation de la consommation d'énergie liée à l'écoute radio.

En ce qui concerne l'utilisation du CPU, nous observons une augmentation due aux calculs supplémentaires requis pour recalculer les routes et gérer les conséquences de l'attaque de modification du numéro de version.

D'autre part, l'attaque de modification du numéro de version peut également perturber les périodes de faible consommation d'énergie (LPM), en raison des changements dans le réseau, ce qui entraîne une augmentation de la consommation d'énergie pendant ces périodes.

2. **Messages de contrôle**

Message de contrôle	Sans attaque	Avec attaque
DIS	14	14
DIO	258	665
DAO	99	1045

TAB. 4.7 : le nombre de message de contrôle avant et après l'attaque Version number.

En se référant du tableai ci-dessus nous pouvons observer que la surcharge de contrôle dans le scénario normal, où une réparation globale est déclenchée, augmente. Cela s'explique par le fait que les nœuds du réseau diffusent des messages DIO pour reconstruire le graphe DODAG.

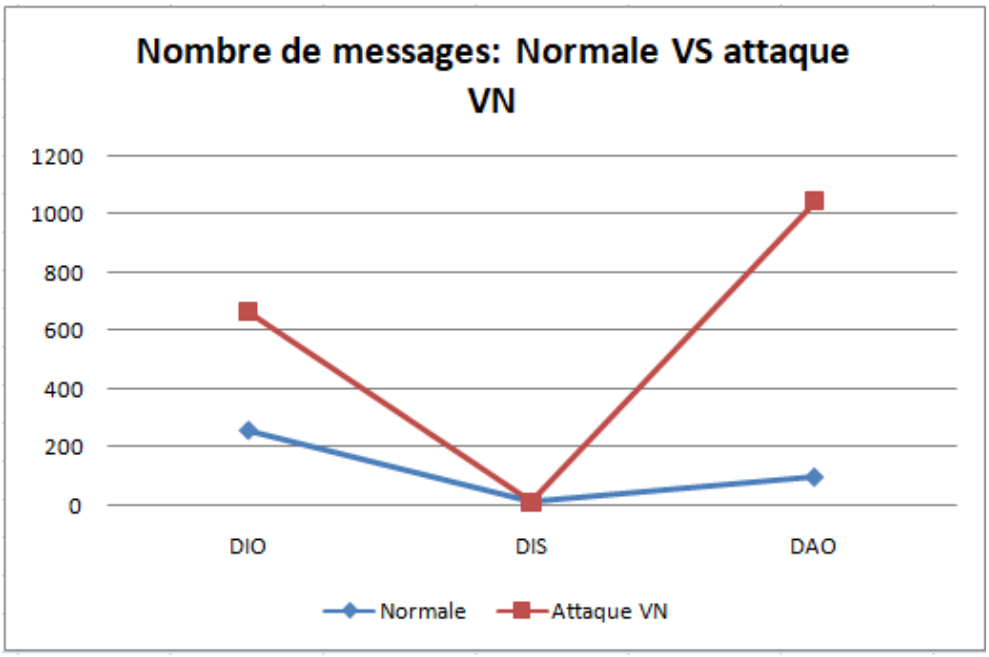

Fig. 4.20 : La courbe des messages de contrôle pendant l'attaque numéro de version et sans attaque.

En revanche, la présence du nœud malveillant 15 lors de l'attaque par numéro de version, entraîne une augmentation significative de la surcharge de contrôle, passant de 258 à 665 messages DIO, et de 99 à 1045 DAO, en raison des manipulations effectuées sur les numéros de version des messages DIO.

4.7 Conclusion

Dans ce chapitre, nous avons mis en œuvre et simulé quatre attaques, puis analysé leurs effets en utilisant deux métriques principales : l'énergie et le nombre de messages de contrôle. Au terme de notre analyse, nous avons constaté que les attaques Blackhole et Sinkhole sont des attaques de type topologie, car elles ont un impact significatif sur la topologie du réseau ainsi que sur la consommation d'énergie. D'autre part, les attaques DIS (DODAG Information Solicitation) et Numéro de version sont des attaques de type ressources, car elles consomment de l'énergie et provoquent une surcharge du trafic dans le réseau. Toutes ces attaques ont pour effet de déstabiliser le réseau.

Nous avons mené une discussion détaillée sur chaque attaque, en explorant les mécanismes et les conséquences spécifiques de chacune d'entre elles. Cela nous a permis de mieux comprendre les risques et les impacts potentiels de ces attaques sur le fonctionnement et la sécurité du réseau.

Conclusion et perspectives

Conclusion générale

L'internet des objets (IoT) est un réseau d'apareils parmi lesquels les capteurs, les appareils mobiles etc. qui peuvent se connecter à internet. Les LLN sont un autre genre de réseaux sans fils et filaires. Limités reliés par des liaisons instables. Effectivement RPL a été concu comme un protocole de routage efficace et évolutif pour les LLN. En effet le manque de ressources de ces types de réseaux le rend particulièrement vulnérable aux menaces de sécurité.

Par conséquent dans ce mémoire, nous avons présenté quelques types d'attaques, parmi une quinzaine d'attaques décrites, qui agissent sur le réseau RPL à savoir le dis Flooding, le Blackhole, le Sinkhole et Version Number. Nous avons analysé l'impact de ces quatre attaques sur deux métriques du réseau RPL à savoir la consommation d'énergie et le nombre de messages de contrôle, et nous avons constaté que chaque attaque déstabilisait le routage soit en agissant sur les ressources c'est-à-dire l'énergie dans notre cas c'est le DIS Flooding et Version number, et en agissant sur la topologie pour le cas de Blackhole et Sinkhole.

Pour pouvoir faire l'implémentation de ces attaques nous avons eu recours au simulateur COOJA sous Contiki OS. Nous avons lancé des simulations de 10 minutes pour chacune des attaques et d'autres encore de 15 minutes sans les attaques pour justement à travers les deux métriques faire l'analyse et déduire leur impact sur les ressources et la topologie d'un réseau RPL. Ensuite nous avons présenté et analysé les résultats obtenus.

Cette étude a révélé des résultats significatifs. Nous avons pu observer les impacts de ces attaques sur deux métriques clés, à savoir la consommation d'énergie et le nombre de messages de contrôle transmis dans le réseau. En ce qui concerne la consommation d'énergie, nous avons constaté une augmentation significative due aux attaques, ce qui entraîne une utilisation accrue de la batterie des nœuds. Cela peut avoir des conséquences néfastes sur la durée de vie des nœuds et la stabilité globale du réseau. En ce qui concerne le nombre de messages de contrôle, nous avons observé une augmentation notable lors de l'exécution des attaques. Cela peut entraîner une surcharge du réseau, une utilisation inefficace des ressources et une diminution des performances globales.

Ces résultats mettent en évidence la vulnérabilité du protocole RPL face à ces attaques spécifiques et soulignent l'importance de renforcer la sécurité dans les réseaux RPL. Des mesures de prévention et de détection des attaques doivent être mises en place pour préserver l'intégrité et la stabilité du réseau.

La réalisation de ce projet de fin d'étude nous a été très bénéfique où on a pu :

- Approfondir nos connaissances sur un nouveau type de réseaux L'IoT et les LLNs.
- surtout à comprendre le fonctionnement de protocole de routage RPL et ses vulnérabilités
- Apprendre à programmer dans l'environnement de Contiki et maitriser un nouveau simulateur Cooja.

Perspectives

Comme perspective de ce travail, nous envisageons de rajouter un autre critère d'évaluation.

Le thème traité est un thème de recherche qui nécessite une continuité, donc dans le futur, nous prévoyons de travailler sur l'exploration de nouvelles méthodes d'amélioration et pourquoi pas de développer des contre-mesures efficaces pour renforcer la sécurité des réseaux IoT basés sur RPL qui permettent de contrer ces attaques.

Bibliographie

[8] Younes Abbassi et Habib Benlahmer. "Un aperçu sur la sécurité de l'internet des objets (IOT)". In : *Colloque sur les Objets et systèmes Connectés-COC'2021*. 2021.

[9] David Airehrour, Jairo Gutierrez et Sayan Kumar Ray. "Securing RPL routing protocol from blackhole attacks using a trust-based mechanism". In : *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE. 2016, p. 115-120.

[10] Ghada Aljufair, Mohammed Mahyoub et Abdulaziz S Almazyad. "On Mitigating DIS Attacks in IoT Networks". In : *2023 18th Wireless On-Demand Network Systems and Services Conference (WONS)*. IEEE. 2023, p. 104-109.

[11] Ioannis Andrea, Chrysostomos Chrysostomou et George Hadjichristofi. "Internet of Things : Security vulnerabilities and challenges". In : *2015 IEEE symposium on computers and communication (ISCC)*. IEEE. 2015, p. 180-187.

[12] Naceur BELHADJ et Abdelhak ABBAD. *La sécurité de l'Internet des Objets (IoT)*. http://dspace.univ-tiaret.dz:80/handle/123456789/2580. 2022.

[13] Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. "Synthèse sur les protocoles de communication pour l'Internet des objets de l'industrie 4.0". Thèse de doct. LS2N, Université de Nantes ; IETR, Université de Nantes, 2019.

[14] Mohammed Amine Boudouaia et al. "RPL rank based-attack mitigation scheme in IoT environment". In : *International Journal of Communication Systems* 34.13 (2021), e4917.

[15] Mohammed Amine Boudouaia et al. "Security against rank attack in RPL protocol". In : *IEEE Network* 34.4 (2020), p. 133-139.

[16] Hadjer BOUZEBIBA. "Adaptation des protocoles de communication conçus pour les IoT aux systèmes IoMT". Thèse de doct. 08-04-2021.

[17] Muhammad Burhan et al. "IoT Elements, Layered Architectures and Security Issues : A Comprehensive Survey". In : *Sensors* 18 (2018).

[18] Muhammad Burhan et al. "IoT elements, layered architectures and security issues : A comprehensive survey". In : *sensors* 18.9 (2018), p. 2796.

[19] Abiy Biru Chebudie, Roberto Minerva et Domenico Rotondi. "Towards a definition of the Internet of Things (IoT)". Thèse de doct. Août 2014.

[20] Khalid A Darabkh et al. “RPL routing protocol over IoT : A comprehensive survey, recent advances, insights, bibliometric analysis, recommendations, and future directions”. In : *Journal of Network and Computer Applications* (2022), p. 103476.

[21] Jasenka Dizdarević et al. “A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration”. In : *ACM Computing Surveys (CSUR)* 51.6 (2019), p. 1-29.

[22] Anuj Kumar Dwivedi et Manoj Kumar Tiwari. “Operating Systems for Tiny Networked Sensors : A Survey”. In : *International Journal of Recent Trends in Engineering (IJRTE)* 1 (mai 2009), p. 152-157.

[23] Hicham El Mrabet et ait moussa Abdelaziz. *L'INTERNET DES OBJETS ET LES TICE : Vers une école intelligente*. Mai 2017.

[24] Achraf Fayad. “Secure authentication protocol for Internet of Things”. Thèse de doct. Institut Polytechnique de Paris, 2020.

[25] Olfa Gaddour et Anis Koubâa. “RPL in a nutshell : A survey”. In : *Computer Networks* 56.14 (2012), p. 3163-3178.

[26] Jorge Granjal, Edmundo Monteiro et Jorge Sá Silva. “Security for the Internet of Things : A Survey of Existing Protocols and Open Research Issues”. In : *IEEE Communications Surveys Tutorials* 17 (juil. 2015), p. 1-1.

[27] A Houha, Siham Mehah, Lamia Ouabba et al. “Internet of Things, protocoles de communication et simulation d'un scénario [maison intelligente].” Thèse de doct. Université Abderrahmane Mira-Bejaia, 2021.

[28] Oana Iova, Fabrice Theoleyre et Thomas Noel. “Using multiparent routing in RPL to increase the stability and the lifetime of the network”. In : *Ad Hoc Networks* 29 (2015), p. 45-62.

[29] Arshad Junaid et al. “A review of performances, energies, and privacies of intrusions detections system for IoTs”. In : *Electronic* 9.629 (2020), p. 1-24.

[30] Patrick Olivier Kamgueu. “Configuration dynamique et routage pour l'internet des objets”. Thèse de doct. Université de Lorraine, 2017.

[31] Harith Kharrufa, Hayder AA Al-Kashoash et Andrew H Kemp. “RPL-based routing protocols in IoT applications : A review”. In : *IEEE Sensors Journal* 19.15 (2019), p. 5952-5967.

[32] A Krari, A Hajami et E Jarmouni. “Study and Analysis of RPL Performance Routing Protocol Under Various Attacks”. In : *International Journal on “Technical and Physical Problems of Engineering”(IJTPE)* 13.49 (2021), p. 152-161.

[33] Arvind Kumar, Rakesh Matam et Shailendra Shukla. “Impact of packet dropping attacks on RPL”. In : *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*. IEEE. 2016, p. 694-698.

[34] Anhtuan Le et al. “The impacts of internal threats towards routing protocol for low power and lossy network performance”. In : *2013 IEEE symposium on computers and communications (ISCC)*. IEEE. 2013, p. 000789-000794.

[35] “Les performances du protocole de routage pour les réseaux à faible consommation et avec perte dans les réseaux mobiles”. In : ().

[36] Diakite Madibaba. “Analyse Et Contre Mesure Des Attaques De Topologie Dans Les Réseaux Lln”. Mém. de mast. Université Abdelhamid Ibn Badis - Mostaganem, 2021.

[37] Avleen Malhi, Shalini Batra et Husanbir Pannu. “Security of Vehicular Ad-hoc Networks : A Comprehensive Survey”. In : *Computers Security* 89 (nov. 2019), p. 101664.

[38] Haniche Malika et Tabrait Nabila. “Internet des objets dans le domaine de l’agriculture de demain.” Thèse de doct. Université Mouloud Mammeri, 2019.

[39] Smitesh Mangelkar, Sudhir N Dhage et Anant V Nimkar. “A comparative study on RPL attacks and security solutions”. In : *2017 International Conference on Intelligent Computing and Control (I2C2)*. IEEE. 2017, p. 1-6.

[40] Anthéa Mayzaud, Rémi Badonnel et Isabelle Chrisment. “A Taxonomy of Attacks in RPL-based Internet of Things”. In : *International journal of network security* 18.3 (mai 2016), p. 459-473.

[41] Fredrik Osterlind et al. “Cross-Level Sensor Network Simulation with COOJA”. In : *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*. 2006, p. 641-648.

[42] Cong Pu. “Sybil Attack in RPL-Based Internet of Things : Analysis and Defenses”. In : *IEEE Internet of Things Journal* 7.6 (2020), p. 4937-4949.

[43] Rajasekar Ramalingam et Rajkumar Soundrapandiyan. “Analysis of Blackhole Attack in RPL-based 6LoWPAN Network : A Case Study”. In : nov. 2021, p. 1-6.

[44] Anass Rghioui, Anass Khannous et Mohammed Bouhorma. “Denial-of-Service attacks on 6LoWPAN-RPL networks : Threats and an intrusion detection system proposition”. In : *Journal of Advanced Computer Science & Technology* 3.2 (2014), p. 143.

[45] Mehdi Rouissat, Mohammed Belkheir et Hichem Sid Ahmed Belkhira. “A potential flooding version number attack against RPL based IOT networks”. In : *Journal of Electrical Engineering* 73.4 (2022), p. 267-275.

[46] Tara Salman et Raj Jain. “A survey of protocols and standards for internet of things”. In : *arXiv preprint arXiv :1903.11549* (2019).

[47] Shadi Al-Sarawi et al. “Internet of Things (IoT) communication protocols”. In : *2017 8th International conference on information technology (ICIT)*. IEEE. 2017, p. 685-690.

[48] Alparslan Sari, Alexios Lekidis et Ismail Butun. “Industrial networks and IIoT : Now and future trends”. In : *Industrial IoT : Challenges, Design Principles, Applications, and Security* (2020), p. 3-55.

[49] Pallavi Sethi et Smruti R Sarangi. “Internet of things : architectures, protocols, and applications”. In : *Journal of Electrical and Computer Engineering* 2017 (2017).

[50] Divya Sharma, Ishani Mishra et Sanjay Jain. “A detailed classification of routing attacks against RPL in internet of things”. In : *International Journal of Advance Research, Ideas and Innovations in Technology* 3.1 (2017), p. 692-703.

[51] Hsu Tsai et Lin Lin. “Tsai CF, Hsu YF, Lin CY, Lin WY”. In : *Intrusion detection by machine learning : a review, Expert Syst. Appl* 36.10 (2009), p. 11994-12000.

[52] Abhishek Verma et Virender fileRanga. “Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks”. In : *Transactions on emerging telecommunications technologies* 31.2 (2020), e3802.

[53] Abhishek Verma et Virender Ranga. “Analysis of routing attacks on RPL based 6LoWPAN networks”. In : *International Journal of Grid and Distributed Computing* 11.8 (2018), p. 43-56.

[54] Linus Wallgren, Shahid Raza et Thiemo Voigt. “Research Article Routing Attacks and Countermeasures in the RPL-Based Internet of Things”. In : *Hindawi Publishing Corporation International Journal of Distributed Sensor Networks Volume* (2013).

[55] Tim Winter et al. *RPL : IPv6 routing protocol for low-power and lossy networks*. Rapp. tech. 2012.

[56] Furkan Yusuf Yavuz, ÜNAL Devrim et GÜL Ensar. “Deep learning for detection of routing attacks in the internet of things”. In : *International Journal of Computational Intelligence Systems* 12.1 (2018), p. 39.

Webographie

[1] URL : https://www.12h15.fr/notice-quest-ce-quun-objet-connecte/ (visité le 18/02/2023).

[2] URL : http://visioforce.com/smarthome.html/ (visité le 03/03/2023).

[3] URL : https://www.al-enterprise.com/-/media/assets/internet/documents/iot-for-education-solutionbrief-fr.pdf (visité le 01/04/2023).

[4] URL : https://connect.ed-diamond.com/open-silicium/os-019/mise-en-place-d-un-reseau-iot-avec-riot (visité le 01/04/2023).

[5] URL : https://www.vmware.com/products/fusion/faq.html (visité le 03/05/2023).

[6] URL : https://sourceforge.net/projects/contiki/files/Instant (visité le 15/05/2023).

[7] URL : https://www.mediafire.com/file/frufrnrqbaz50j7/net%5C_set%5C_eng%5C_traf%5C_loss1.pl/file (visité le 01/05/2023).

Annexes

Annexe A

COOJA

L'objectif de cet Annexe est de décrire la mise en œuvre d'un réseau de référence ainsi que la simulation d'attaques à l'aide du simulateur Cooja. Pour cela, différents nœuds malveillants ont été déployés et les attaques ont été réalisées en modifiant les fichiers de configuration RPL, ce qui a entraîné des modifications dans le comportement des nœuds. Ce chapitre se concentre sur l'aperçu de la création du réseau de référence à travers l'interface graphique de Cooja, la mise en œuvre des différentes attaques et la collecte de la consommation d'énergie des nœuds.

A.1 Le réseau initial

Pour démontrer l'impact des attaques sur les nœuds, il sera nécessaire d'obtenir des données de référence pour les comparer. Ainsi, cette section vise à expliquer comment mettre en œuvre un réseau de référence basé sur les paramètres décrits dans le tableau de paramètres de simulation. La construction d'un réseau de référence comprend les étapes suivantes :

1. Pour créer une nouvelle simulation, cliquez sur Fichier -> Nouvelle simulation dans la barre de menu. Ici, vous devrez définir le nom de la simulation et le type de support radio.

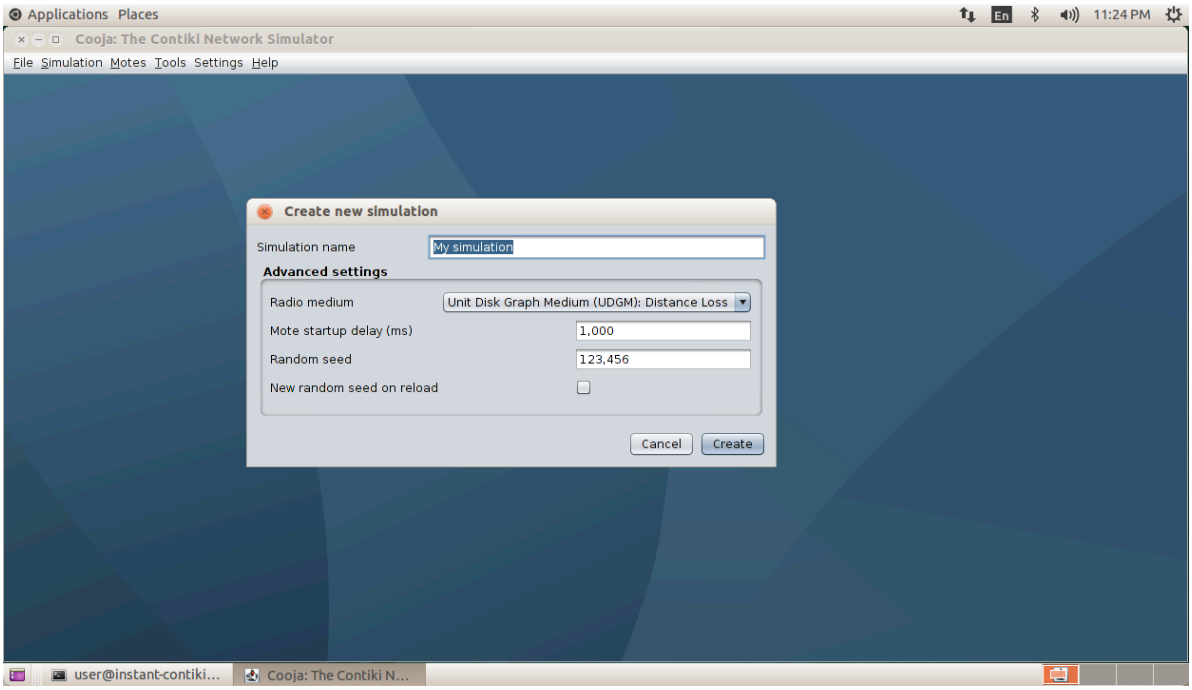

FIG. A.1 : Création d'une nouvelle simulation sur cooja.

2. La prochaine étape consistera à créer les types de nœuds qui constitueront le réseau. Le réseau de référence comprendra deux types de nœuds : un nœud de destination (sink node), qui fonctionnera en tant que routeur DODAG, et des nœuds feuille (leaf motes), qui fonctionneront simplement comme des nœuds clients. En cliquant sur Motes -> Ajouter des nœuds -> Créer un nouveau type de nœud -> Z1 mote dans la barre de menu, une fenêtre apparaîtra. Ici, les firmwares des différents nœuds seront compilés pour les créer

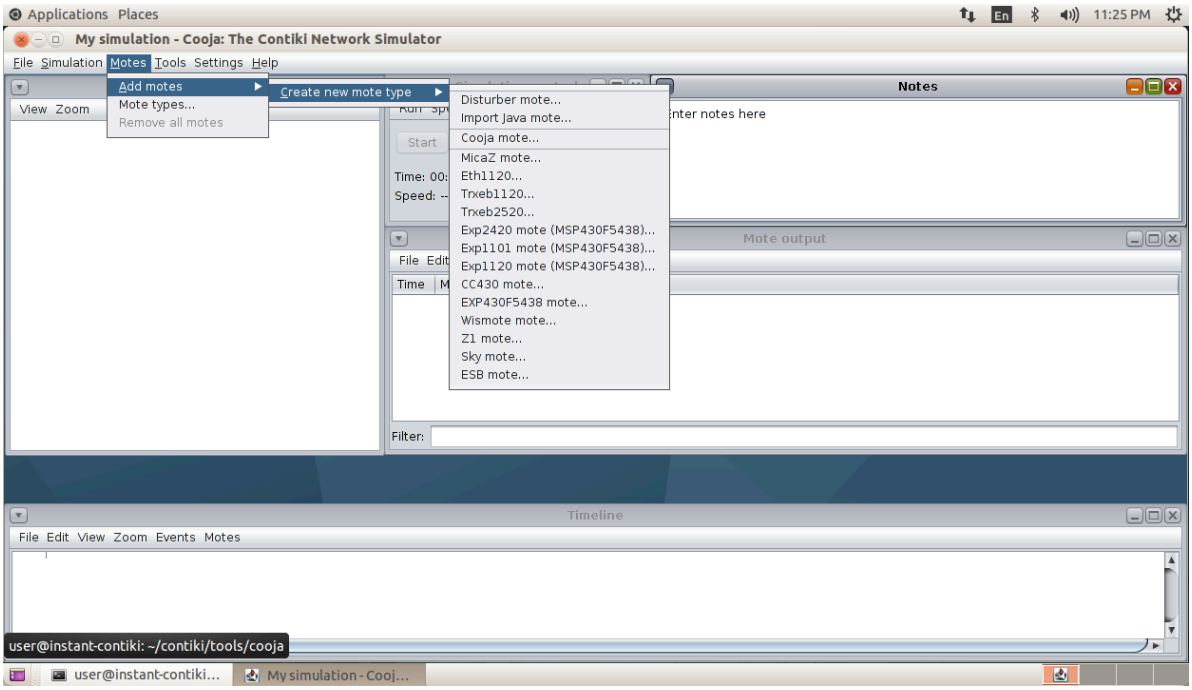

FIG. A.2 : L'ajout des noeuds à la simulation.

3. le nœud root sink est basé sur le fichier firmware suivant : /Contiki/examples/ipv6/rpl-collect/sink.c - sink. Ensuite, appuyer sur le bouton "Compiler" puis "Créer".

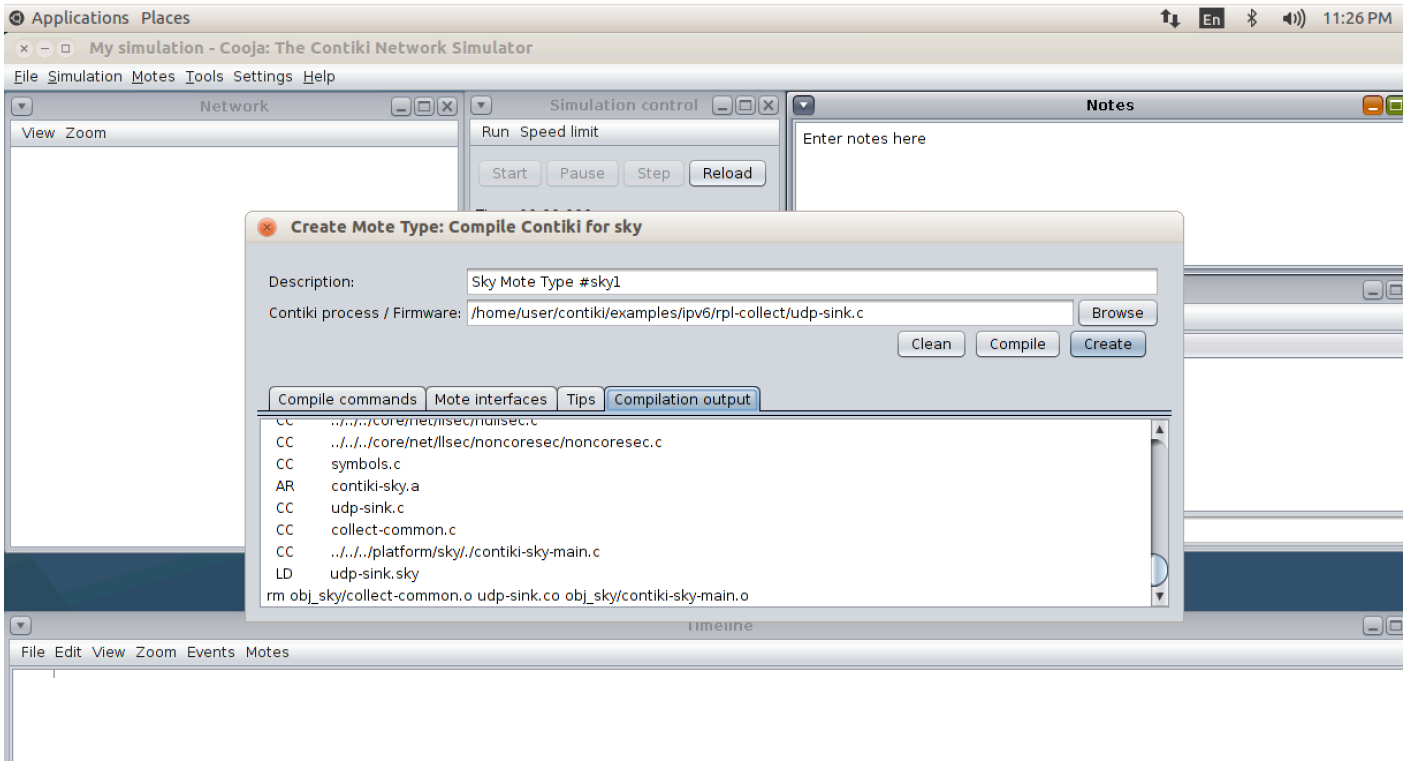

FIG. A.3 : téléchargement de code de sink mote.

4. Une fois que le fichier a été compilé avec succès le nœud sera crée.

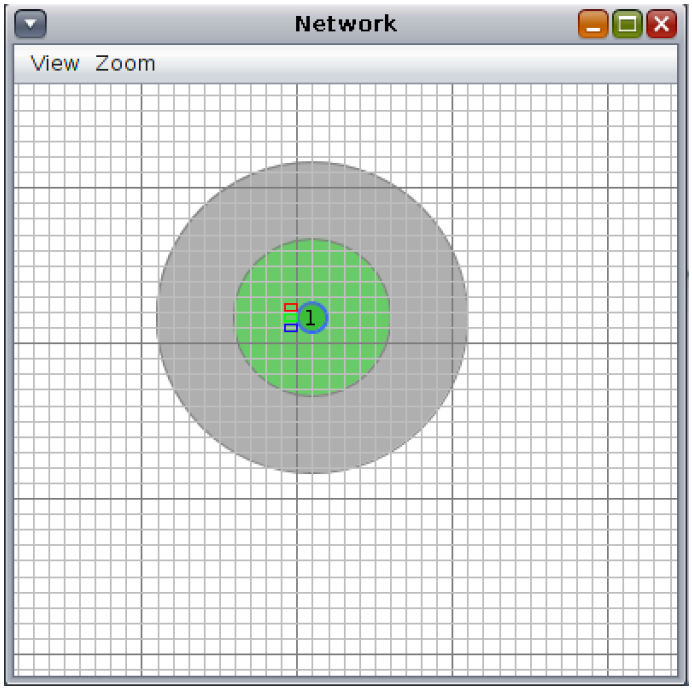

FIG. A.4 : Le noeud sink (racine) .

5. Pour ajouter les autres nœuds on suit les mêmes étapes mais avec le firmware suivant :
/Contiki/examples/ipv6/rpl-collect/udp-sender.c - motes.

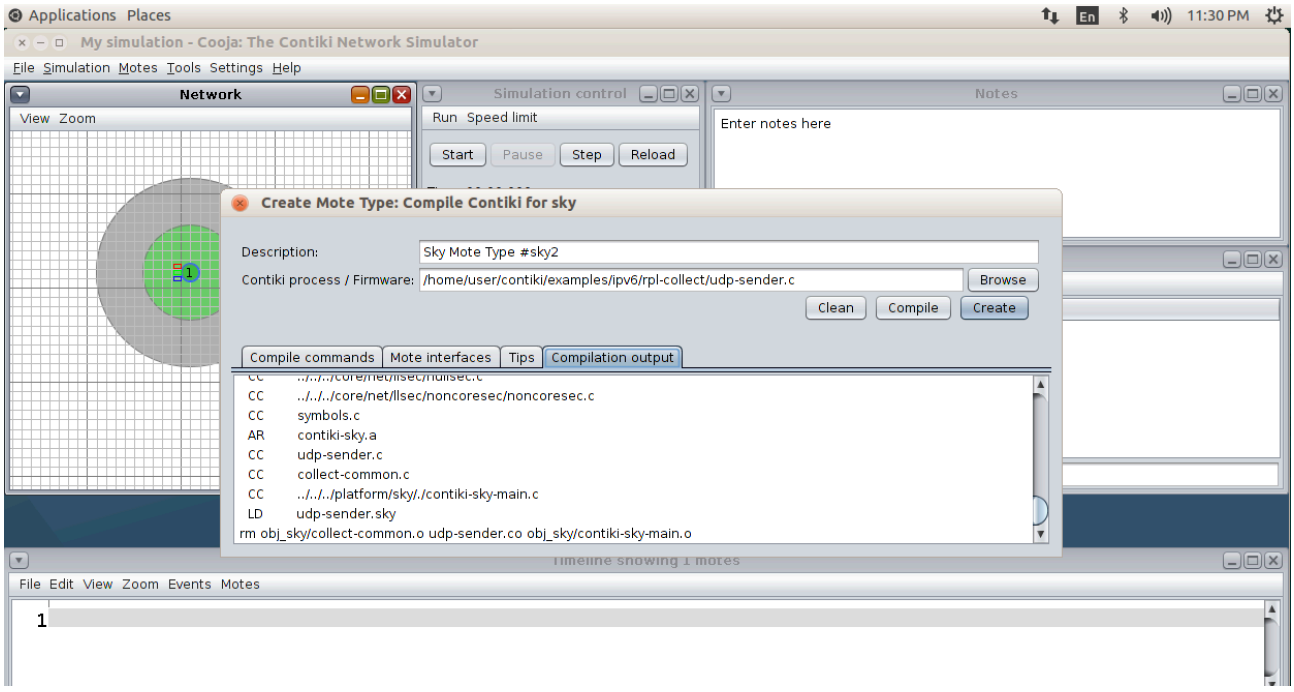

FIG. A.5 : L'ajout des noeuds clients.

6. L'environnement de simulation final est représenté dans la figure xxx . L'image montre également la portée de transmission et d'interférence du routeur DODAG.

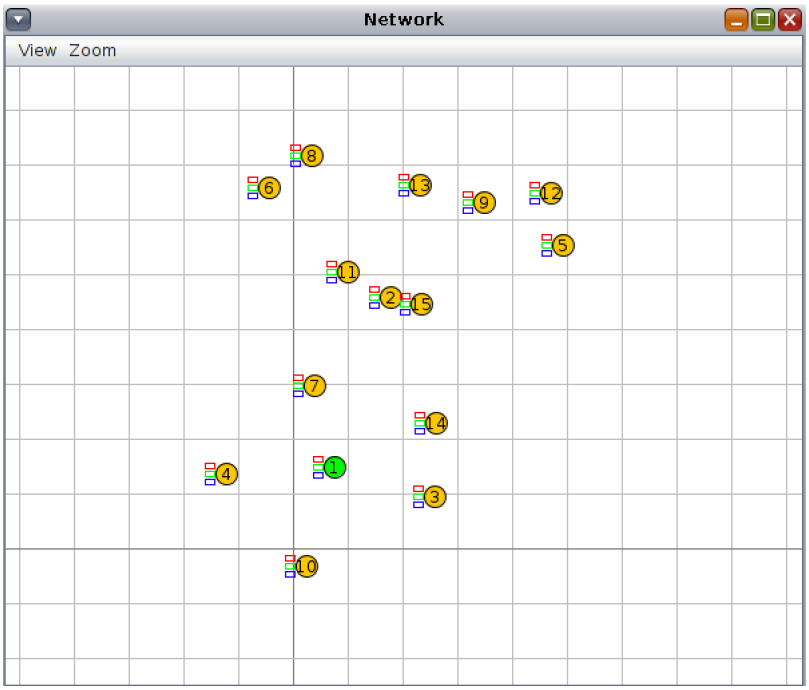

FIG. A.6 : La topologie créée avec l'emplacement des noeuds

7. **Démarrage de simulartion** : le démarrage s'effectue par un simple clic sur Start.

8. **Récupération des données de la simulation** : COOJA offre la possibilité de récupérer les données de la simulation tel que (les messages transmis, le contenu de trafique ...etc.) dans un fichier en utilisant le fichier généré Cooja.testlog.

Une fois le réseau de référence créé, Les nœuds sont ajoutés à la zone de simulation, comme on peut le voir. En cliquant sur l'un des nœuds, la portée de chaque nœud s'affiche. Le rayon vert représente une portée valide dans laquelle les nœuds peuvent communiquer directement les uns avec les autres. En dehors de cette portée, ils ne pourront pas communiquer ou nécessiteront plusieurs sauts en fonction du protocole utilisé.il sera possible de collecter les données qui serviront de base à l'étude. Les sections suivantes décriront comment mettre en œuvre des nœuds malveillants au sein du réseau de référence et déclencher des attaques.

A.2 Implémentation des attaques

A.2.1 L'objectif

L'objectif principal lors de la simulation d'une attaque est de modifier le comportement d'un ou plusieurs nœuds sans altérer le comportement normal des autres membres du réseau. Ainsi, il est possible d'évaluer la réaction du réseau face à des situations inhabituelles.

A.2.2 Méthodologie

La méthode utilisée pour atteindre cet objectif peut être réalisée en suivant les étapes suivantes :

- Dupliquer le dossier Contiki pour créer une nouvelle instance du système d'exploitation Contiki, on a créer quatre instances puisque nous allons implémenter 4 attaques,
- Modifier les fichiers correspondants en fonction de chaque attaque,
- Créez un nouveau nœud (malveillant) en compilant le firmware du nœud dans la nouvelle instance de Contiki,
- Ajoutez le nœud au réseau de référence.

A.2.3 L'attaque Blackhole

Pour implémenter l'attaque blackhole, le fichier ''uip6.c'' qui est situé dans l'instante Contiki créée pour cette attaque doit être modifié, le dossier de modification esr situé dans : ''contiki/core/net/ipv6 ''.

Uip6.c :contient l'implémentation de la pile uIP TCP/IPv6 pour Contiki, il comprend le code qui transmet les paquets non destinés à un nœud précis vers leurs destination, pour réaliser l'attaque blackhole le nœud malveillant devrait abandonner tout ces paquets en modifiant le code suivant :

```
  UIP_IP_BUF->ttl = UIP_IP_BUF->ttl - 1;
  PRINTF("Forwarding packet to ");
  PRINT6ADDR(&UIP_IP_BUF->destipaddr);
  PRINTF("\n");
  UIP_STAT(++uip_stat.ip.drop);         //remplacer UIP_STAT(++uip_stat.ip.forwarded) par UIP_STA(+uip_stat.ip.drop) ;
  goto drop;                           //remplacer goto send par goto drop;
} else {
  if((uip_is_addr_link_local(&UIP_IP_BUF->srcipaddr)) &&
     (!uip_is_addr_unspecified(&UIP_IP_BUF->srcipaddr)) &&
     (!uip_is_addr_loopback(&UIP_IP_BUF->destipaddr)) &&
     (!uip_is_addr_mcast(&UIP_IP_BUF->destipaddr)) &&
     (!uip_ds6_is_addr_onlink((&UIP_IP_BUF->destipaddr))))
    PRINTF("LL source address with off link destination, dropping\n");
    uip_icmp6_error_output(ICMP6_DST_UNREACH,
                           ICMP6_DST_UNREACH_NOTNEIGHBOR, 0);
    goto send;
```

FIG. A.7 : La modification pour l'attaque blackhole

A.2.4 L'attaque Sinkhole

L'attaque sinkhole est une combinaison de deux attaques : l'attaque de diminuation du rang (decrease rank attack) et l'attaque du trou noir (blackhole attack). Pour réaliser cette attaque, il sera nécessaire de modifier du code dans les fichiers RPL qui sont situés dans : ''contiki/core/net/rpl/''.

Rpl_private.h : Ce fichier contient les déclarations privées pour l'implémentation RPL

Contiki, comme les valeurs par défaut des messages de contrôle, les temporisateurs, le mode de fonctionnement, les tables de routage DAG et il contient diverses définitions liées au calcul du classement DAG. L'implémentation de l'attaque diminution du rang peut être mise en œuvre en modifiant certaines de ses constantes, la figure suivante présente les modifications du code.

```
#define RPL_LIFETIME(instance, lifetime) \
         ((unsigned long)(instance)->lifetime_unit * (lifetime))

#ifndef RPL_CONF_MIN_HOPRANKINC
#define RPL_CONF_MIN_HOPRANKINC     0                                   //modifier set RPL_CONF_MIN_HOPRANKINC à 0
#define RPL_MIN_HOPRANKINC          256
#else
#define RPL_MIN_HOPRANKINC          RPL_CONF_MIN_HOPRANKINC
#endif
#define RPL_MAX_RANKINC             0                                   //changer (7 * RPL_MIN_HOPRANKINC) à 0

#define DAG_RANK(fixpt_rank, instance) \
  ((fixpt_rank) / (instance)->min_hoprankinc)

/* Rank of a virtual root node that coordinates DAG root nodes. */
#define BASE_RANK                       0

/* Rank of a root node. */
#define ROOT_RANK(instance)             (instance)->min_hoprankinc

#define INFINITE_RANK                   256                             //remplacer 0xffff par 256
```

Fig. A.8 : Les modifications ajoutées pour déclancher l'attaque Diminution du rang.

Rpl_timers.c : Il contient le code qui recalcule les rangs des nœuds en RPL, l'implémentation de l'attaque du rang nécéssite également de désactivver ce recalcul, afin que les effets de diminution du rang ne seront pas annulés, pour cela on supprime la ligne concernée.

```
/*---------------------------------------------------------------------------*/
static void
handle_periodic_timer(void *ptr)
{
  rpl_purge_routes();
  //rpl_recalculate_ranks();                          supprimer cette ligne

  /* handle DIS */
#if RPL_DIS_SEND
  next_dis++;
  if(rpl_get_any_dag() == NULL && next_dis >= RPL_DIS_INTERVAL) {
    next_dis = 0;
    dis_output(NULL);
  }
#endif
  ctimer_reset(&periodic_timer);
}
/*---------------------------------------------------------------------------*/
```

Fig. A.9 : supression de la ligne du code source.

A.2.5 L'attaque DIS Flooding

L'attaque DIS peut être implémentée en modifiant le code source des fichiers suivants qui se trouvent dans ''contiki/core/net/rpl'' :

Rpl_private.h : qui contient plusieurs constantes reliées au fonctionnement des messages DIS, dans ce fichier on modifie la valeur de l'intervalle des messages DIS et le temporisateur de démarrage, nous mettons les deux valeurs à 0 pour un effet maximum.

```
/* DIS related */
#define RPL_DIS_SEND                    1
#ifdef  RPL_DIS_INTERVAL_CONF
#define RPL_DIS_INTERVAL                RPL_DIS_INTERVAL_CONF
#else
#define RPL_DIS_INTERVAL                0 /* la valeur etait 60 */
#endif
#define RPL_DIS_START_DELAY             0 /* la valeur etait 5 */
/*---------------------------------------------------------------------------*/
/* Lollipop counters */

#define RPL_LOLLIPOP_MAX_VALUE           255
#define RPL_LOLLIPOP_CIRCULAR_REGION     127
#define RPL_LOLLIPOP_SEQUENCE_WINDOWS    16
#define RPL_LOLLIPOP_INIT                (RPL_LOLLIPOP_MAX_VALUE - RPL_LOLLIPOP_SEQUENCE_WINDOWS + 1)
#define RPL_LOLLIPOP_INCREMENT(counter)                                    \
```

FIG. A.10 : Modification du fichier Rpl_private.h attaque DIS.

Rpl_timers.c : Une boucle est créée dans la fonction de minuterie périodique pour envoyer des messages DIS sans aucune condition.

```
/*---------------------------------------------------------------------------*/
static void
handle_periodic_timer(void *ptr)
{
  rpl_purge_routes();
  rpl_recalculate_ranks();

  /* handle DIS */
#if RPL_DIS_SEND
  next_dis++;
  int i=0;
  while (i<20) {i++;dis_output(NULL);}
  if(rpl_get_any_dag() == NULL && next_dis >= RPL_DIS_INTERVAL) {
    next_dis = 0;
    dis_output(NULL);
  }
#endif
  ctimer_reset(&periodic_timer);
}
/*---------------------------------------------------------------------------*/
```

FIG. A.11 : Ajouter une boucle dans le fichier Rpl_timers.c .

A.2.6 L'attaque sur le numéro de version

Il est nécéssaire de modifier du code dans le fichier RPL Rpl-icmp6 qui se trouve dans ''contiki/core/net/rpl'', pour forcer un noeud à envoyer des messades DIO avec une version incrémentée.

```
  /* DAG Information Object */
  pos = 0;

  buffer = UIP_ICMP_PAYLOAD;
  buffer[pos++] = instance->instance_id;
  buffer[pos++] = dag->version++; /* modifier version par version++ */

#if RPL_LEAF_ONLY
  PRINTF("RPL: LEAF ONLY DIO rank set to INFINITE_RANK\n");
  set16(buffer, pos, INFINITE_RANK);
#else /* RPL_LEAF_ONLY */
  set16(buffer, pos, dag->rank);
#endif /* RPL_LEAF_ONLY */
  pos += 2;
```

FIG. A.12 : modification de la version du DODAG pour attaque de version.

A.2.7 Création d'un nœud malicieux

La création d'un nœud malveillant se fait dans une autre instance de Contiki. Dans le cadre de l'étude, nous avons remplacé le noeud 15 par un noeud malicieu, son emplacement est différent pour chaque attaque afin de démontrer son impact sur RPL. voici un exemple sur la création du nœud malveillant :

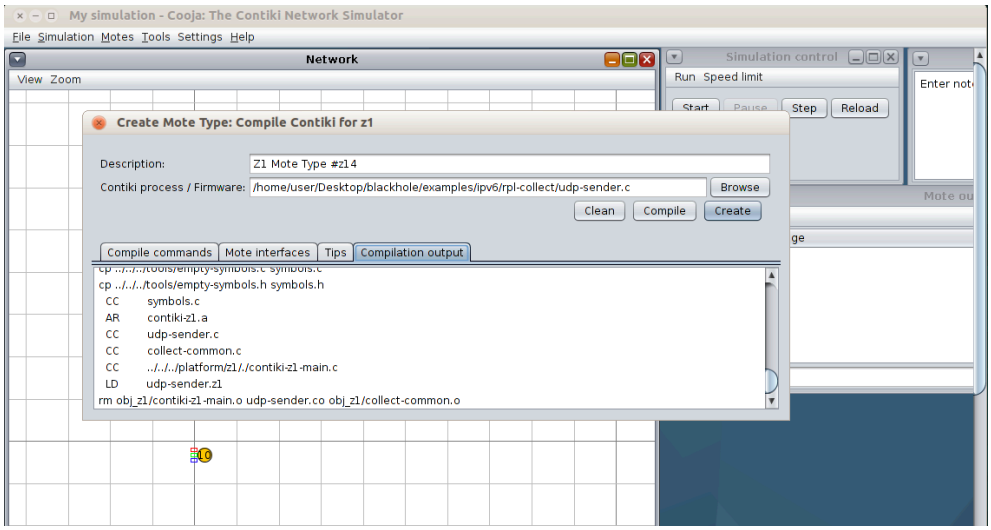

FIG. A.13 : L'ajout du noeud malicieu.

Le réseau avec le nœud malveillant est représenté ci-dessous.

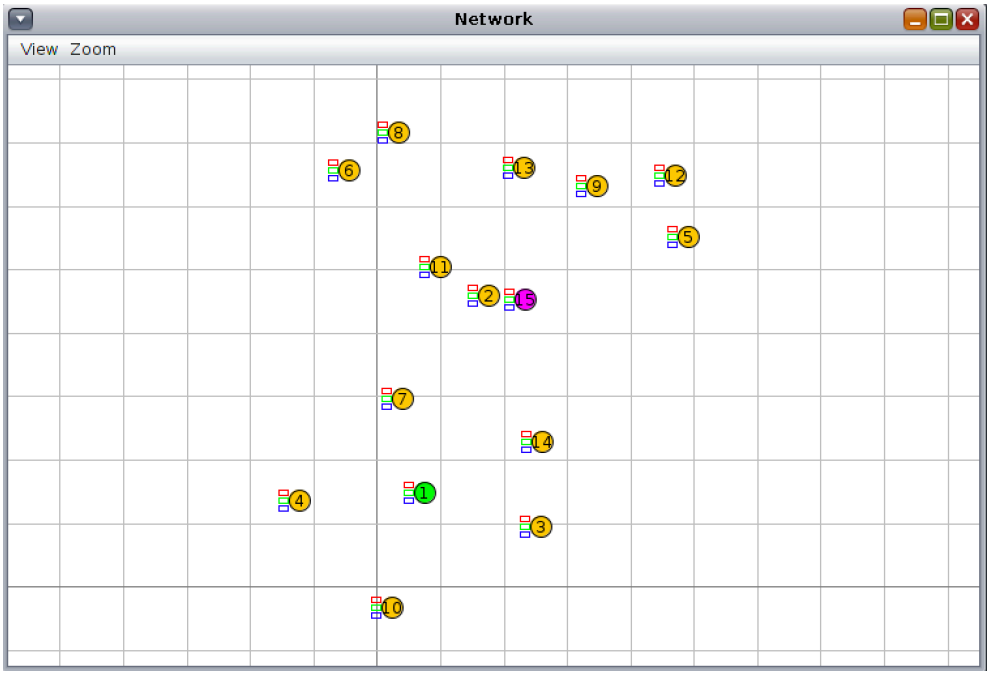

FIG. A.14 : Topologie du réseau avec un noeud malicieu.

A.3 Calcul du nombre des messages de controle

A l'aide d'un script perl téléchargé depuis [7].

Résumé

L'internet des objets (IoT) est un domaine en constante expansion, où les dispositifs connectés sont utilisés pour collecter et transmettre des données. Le protocole RPL est souvent utilisé dans les réseaux de capteurs sans fil IoT pour assurer une communication efficace entre les dispositifs. Cependant, l'utilisation du protocole RPL dans les réseaux IoT peut également présenter des risques de sécurité.

Dans ce projet, nous nous intéressons à la question de sécurité et au fonctionnement de RPL dans un premier temps. Ensuite nous allons faire un état de l'art sur les travaux existants qui étudient les attaques RPL et leur classification, puis nous allons implémenter quatre attaques (Backhole, Sinkhole, DIS et Version number) à l'aide du simulateur Cooja pour analyser les conséquences des attaques sur les paramètres tels que la consommation d'énergie et la surcharge du trafic de contrôle pour analyser leurs impacts sur le réseau et le fonctionnement du protocole de routage RPL, enfin nous discuterons les résultats obtenus.

Mots-clés : Internet des Objet, RPL, LLN, Cooja, Simulation, Attaques, sécurité.

Abstract

The Internet of Things (IoT) is a rapidly expanding field, where connected devices are used to collect and transmit data. The RPL protocol is often used in IoT wireless sensor networks to ensure efficient communication between devices. However, the use of the RPL protocol in IoT networks can also introduce security risks.

In this project, we focus on the security aspects and the operation of RPL. Firstly, we examine the functioning of RPL and its security implications. Next, we conduct a comprehensive review of existing research that investigates RPL attacks and their classification. We then proceed to implement four attacks (Blackhole, Sinkhole, DIS, and Version number) using the Cooja simulator to analyze the impact of these attacks on parameters such as energy consumption and control traffic overhead, in order to assess their effects on the network. Finally, we discuss the obtained results.

Keywords: Internet of Things, RPL, LLN , Simulation, Attacks, Cooja, security.