REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunication

Par :

- Asnoune Ismail
- Ammour Mohamed

Sur le thème

La détection des attaques DoS en utilisant l'analyse de la variance multivariée MANOVA dans les réseaux de capteurs sans fil

---

Soutenu le 15/ 11 / 2020 à Tiaret devant le jury composé de :

Mr Benghani Abdelmalek	Grade Université MAA	Président
Mr BOUALEM Adda	Grade Université MAA	Examinateur
Mr BEKKAR Khaled	Grade Université MAA	Encadreur

Remerciements

Nous remercions avant tout ALLAH qui nous a permis d'arriver jusque là

Paix et salut sur notre premier éducateur le prophète

Nous tenons à présenter respectueusement notre sincère gratitude à

MONSIEUR / BEEKAR KHALED notre encadreur qui nous a

Aider à réaliser ce travail en nous prodiguant des conseils, des orientations et des

observations aussi bénéfiques que fructueuses les unes que les autres.

Nous tenons aussi à remercier les jurés d'avoir accepté de juger notre travail et de

nous honorer de leur présence.

Dédicace

C'est avec profonde gratitude et sincères mots, que nous dédions ce modeste travail de fin d'étude à nos chers parents ; qui ont sacrifié leur vie pour notre réussite et nous ont éclairé le chemin par leurs conseils judicieux.

Nous espérons qu'un jour, nous pourrons leurs rendre un peu de ce qu'ils ont fait pour nous, que dieu leur prête bonheur et longue vie.

Nous dédions aussi ce travail à nos frères et sœurs, nos familles, nos amis, tous nos professeurs qui nous ont enseigné et à tous ceux qui nous sont chers.

ملخص

أصبحت شبكات الاستشعار اللاسلكية واحدة من مجالات البحث الحالية وأثبتت أنها تقنية مفيدة للغاية لمختلف التطبيقات مثل التطبيقات البيئية والعسكرية والصحية والمحلية.

نظرًا لأن شبكات الاستشعار اللاسلكية غالبًا ما تعالج البيانات الحساسة ، وتعمل في بيئات معادية وغير متوقعة ، فقد تقع شبكات الاستشعار اللاسلكية ضحية لهجمات متعددة. يعتبر هجوم رفض الخدمة (DoS) من أخطر الهجمات ، حيث يمكن أن يكون له آثار سلبية على التطبيقات الهامة لـشبكات الإستشعار اللاسلكية، لذلك يعتبر مفهوم الأمن ضروريًا. ومع ذلك ، نظرًا لمحدودية الموارد والقدرة الحسابية المنخفضة ، فإن تطوير آلية لضمان الأمان يطرح تحديات تصميم حقيقية. من بين الحلول المقترحة ، أثبتت أنظمة كشف التسلل (IDS) المستندة إلى النماذج الإحصائية فعاليتها.

في هذا العمل ، درسنا احتمالات اكتشاف هجمات DoS باستخدام تحليل التباين متعدد المتغيرات (MANOVA) بين الحالات العادية والحالات غير الطبيعية.

قمنا بمحاكاة بعض الهجمات Blackhole و Hello-Flood و DoS ، وعلى أساس 30 محاكاة ، فإن أداء نظامنا مقبول نسبيا ، وصلنا إلى دقة 68٪ في سيناريو Blackhole ، ودقة 72٪ في سيناريو Hello-Flood ، ودقة 77٪ في سيناريو DoS.

الكلمات المفتاحية: شبكات الاستشعار اللاسلكية (RCSF) ، نظام كشف التطفل (IDS) ، المحاكاة ، رفض الخدمة (DoS) ، الثقب الأسود (Blackhole)، Hello-Flood،.تحليل التباين متعدد المتغيرات.

RÉSUMÉ

Les réseaux de capteurs sans fil (RCSF) sont devenus l'un des domaines de recherche actuels et se révèlent être une technologie très utile pour diverses applications telles que les applications environnementales, militaires, sanitaires, domestiques .

Le fait que les réseaux de capteurs sans fil traitent des données très souvent sensibles, opérant dans des environnements hostiles et inattendus, les réseaux de capteurs sans fil (RCSF) peuvent être victimes de plusieurs attaques. L'attaque déni de service(DoS) est l'une des plus dangereuses, car ils peuvent avoir des impacts négatifs sur les applications critiques des RCSF, donc la notion de sécurité est considérée comme indispensable. Cependant, à cause de la limitation des ressources et la faible capacité de calcul d'un nœud capteur, le développement d'un mécanisme garantissant une sécurité pose de vrais défis de conception. Parmi les solutions proposées, les systèmes de détection d'intrusion (IDS) basés sur les modèles statistiques ont prouvé leur efficacité.

Dans ce travail, nous avons étudié les possibilités de détecter les attaques DoS en L'analyse de variance multi variée (MANOVA) entre les cas normaux et les cas anomalies.

Nous avons simulé quelques attaques : Blackhole, Hello-Flood et DoS, et sur la base de 30 simulations , les performances de notre système sont plutôt acceptables, nous avons atteint 68% de précision dans le scénario de Blackhole, et 72% de précision dans le scenario Hello-Flood, et 77% de précision dans le scenario DoS.

Mots-clés: Réseaux de capteurs sans fil (RCSF), Système de détection d'intrusion (IDS), Simulations, déni de service (DoS), Blackhole, Hello-Flood, MANOVA.

Abstract

Wireless Sensor Networks (WSN) have become one of the current research fields and are proving to be a very useful technology for various applications such as environmental, military, health, domestic applications.

Because wireless sensor networks process very often sensitive data, operating in hostile and unexpected environments, Wireless Sensor Networks (WSN) can fall victim to multiple attacks. The denial of service (DoS) attack is one of the most dangerous, as it can have negative impacts on critical applications of WSN, so the notion of security is considered essential. However, due to resource limitations and the low computational capacity of a sensor node, developing a mechanism to ensure security. Among the solutions proposed, intrusion detection systems (IDS) based on statistical models have proven their effectiveness.

In this work, we studied the possibilities of detecting DoS attacks using multivariate analysis of variance (MANOVA) between normal cases and anomaly cases.

We simulated a few attacks: Blackhole, Hello-Flood and DoS, and on the basis of 30 simulations , the performance of our system is quite acceptable, we reached 68% accuracy in the Blackhole scenario, and 72% accuracy in the Hello-Flood scenario, and 77% accuracy in the DoS scenario.

Keywords: Wireless Sensor Networks (WSN), Intrusion Detection System (IDS), Simulations, Denial of Service (DoS), Blackhole, Hello-Flood, MANOVA.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS

A

ANOVA: Analysis Of Variance

D

DoS : Denial of Service, « déni de service » en français

H

H-IDS: Host Based Intrusion Detection System

I

IDS: Intrusion Detection System

N

N-IDS: Network Based Intrusion Detection System

M

MANOVA: multivariate analysis of variance (en français: analyse multivarié de variance)

R

RCSF: Réseau de Capteur Sans Fil

S

SMP: Sensor Management Protocol

T

TCP: Transmission Control Protocol

U

UDP: User Datagram Protocol

W

WSN: wireless sensor network

Introduction générale:

Les réseaux de capteurs sans fil (RCSF) ou bien le (WSN) en anglais sont composés de petits appareils qui s'appellent « Capteurs » équipés d'un protocole de communication sans fil. Ces appareils sont mis dans un environnement à étudier, le rôle de ces capteurs c'est la détection des mesures physiques, de les convertir en un signal numérique, et de les transmettre a une station de base pour un traitement plus approfondi, cette station est une interface entre le réseau (RCSF) et l'utilisateur. La collection de ces données à partir le réseau est relié aux contraintes de ressources des capteurs qui sont limitées (la capacité de calcules et de mémoire et l'énergie disponible sur la batterie de capteur).

Cette technologie elle a plusieurs domaines d'applications comme le domaine militaire, l'environnement, la santé, les bâtiments, le transport et le domaine médical.

Les réseaux RCSF sont vulnérables à plusieurs attaques .Certaines attaques ont pour buts l'arrêt total ou partiel du réseau WSN.C'est pour cela la protection de ces réseaux est très nécessaires pour un fonctionnement normal, dont ou Il y a pas mal de mécanismes pour la lute contre les attaques dans RCSF.

Notre mémoire se repose sur la détection des attaques Dos avec un IDS base sur modèle

statistique qui s'appelle l'analyse multi variée de la variance (MANOVA).

Le reste de rapport est organisée comme suit :

-Dans le chapitre 1, nous introduisons l'état de l'art des réseaux de capteurs sans fil, la on a détaillée c'est quoi un capteur et sa composition et après on donne une définition sur les réseaux RCSF avec leurs domaines d'application et des architectures de communication de ces réseaux, enfin nous avons décrire quelque système d'opération pour cette nouveau technologie.

-Dans le chapitre 2, nous avons donné une généralité sur la sécurité des réseaux de capteur sans fil, on a parlé sur les buts de sécurité en suite nous citons les obstacles de sécurité liés aux réseaux de capteurs sans fil, après on donne une définition sur les IDS et leurs types et les différents types des intrusions, en fin on a cité quelque attaques communes dans les réseaux RCSF.

-Dans le dernier chapitre, allons expliquer notre solution en détaille qui est un système IDS basé sur l'analyse multi varié.

CHAPITRE 01

Généralités sur les réseaux de capteurs sans fil

1 Introduction

Les détecteurs avait un rôle simple dans la détection de la chaleur, de l'humidité, des vibrations .Avec les progrès technologiques et les réseaux sans fil continuent de réussir, grâce à ses avantages multiples et uniques qui peuvent se résumer dans la facilité de diffusion de l'information, la propagation de l'information partout et le faible coût d'installation .

À la suite de l'interconnexion de deux pôles de l'informatique moderne les systèmes embarqués et communications sans fil ,sortie une nouvelle technologie appelée réseaux de capteurs sans fil , avec des caractéristiques uniques.

Un réseau de capteurs sans fil (RCSF), ou "Wireless Sensor Network" (WSN), est composé d'un ensemble des nœuds, communiquant via des liens sans fil. Le but général d'un WSN est la collecte d'un ensemble de paramètres de l'environnement entourant les nœuds, telles que la température ou la pression de l'atmosphère, afin de les acheminer vers des points de traitement.

2 Historique des réseaux de capteurs sans fil

Les récents progrès des nouvelles techniques ont provoqué une énorme importance dans le domaine des réseaux sans fil. La technologie des réseaux de capteurs sans fil est devenue une des merveilleuses technologies dans le 21 éme siècle, les réseaux de capteurs ont montré leur impact sur notre vie quotidienne. Le tableau suivant illustre l'évaluation des réseaux de capteur

Génération	Période	Taille	Poids	Batterie
1ère	Les années 80 et 90	Grande boite à chaussure	Kilogrammes	Grosse
2ème	Entre 2000 et 2003	Boite de carte	Grammes	AA
3ème	2010	Particule de poussière	Négligeables	Solaire

Tableau I. 1: Les générations des nœuds de capteurs

3 Réseaux de capteurs sans-fil (RCSF)

3.1 Qu'est ce qu'un capteur ?:

Un capteur est un périphérique, un module, une machine ou un sous-système dont le but est de détecter des événements ou des changements dans son environnement et d'envoyer les informations à d'autres appareils électroniques, Un capteur est toujours utilisé avec d'autres appareils électroniques.

3.2 Architecture physique d'un capteur:

Un capteur est composé de 3 unités :

3.2.1 L'unité d'acquisition :

l'unité d'acquisition est composée d'un capteur qui va obtenir des mesures numériques sur les paramètres environnementaux et d'un convertisseur Analogique/Numérique qui va convertir l'information relevée et la transmettre à l'unité de traitement.

3.2.2 L'unité de traitement :

l'unité de traitement est composée de deux interfaces, une interface pour l'unité d'acquisition et une interface pour L'unité de transmission. Cette unité est également composée d'un processeur et d'un système d'exploitation spécifique. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission.

3.2.3 L'unité de transmission :

l'unité de transmission est responsable de toutes les émissions et réceptions de données via un support de communication radio. [1]

Ces trois unités sont alimentées par une batterie comme le montre la figure ci-dessous :

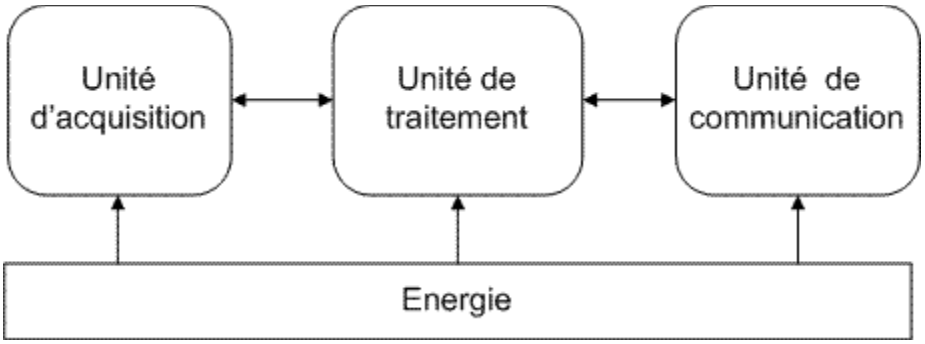


Figure I. 1 Architecture physique d'un capteur [2]

3.3 Définition d'un Réseau capteur sans fil :

Un réseau de capteurs sans fil est un réseau ad hoc avec la plupart de nœuds qui sont des micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome. La position de ces nœuds n'est pas obligatoirement déterminée. Ils peuvent être aléatoirement dispersés dans une zone géographique, nommée «champ de captage» correspondant au terrain d'intérêt pour le phénomène capté.

3.4 Composition d un Réseau capteur sans fil :

Un réseau de capteurs est composé de deux types de nœuds :

- les capteurs.

- le(s) puits.

Les capteurs sont chargés de relever et de router les informations relevées sur la zone couverte vers le point de collecte, également appelé puits.

Le puits récupère les informations remontées par les différents capteurs et les transmet au centre de traitement. Les capteurs disposés de manière aléatoire forment la zone de couverture. [3]

3.5 Architecture de communication:

Il y a deux méthodes pour communication et collecter les informations d'un réseau de capteurs :

3.5.1 A la demande :

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

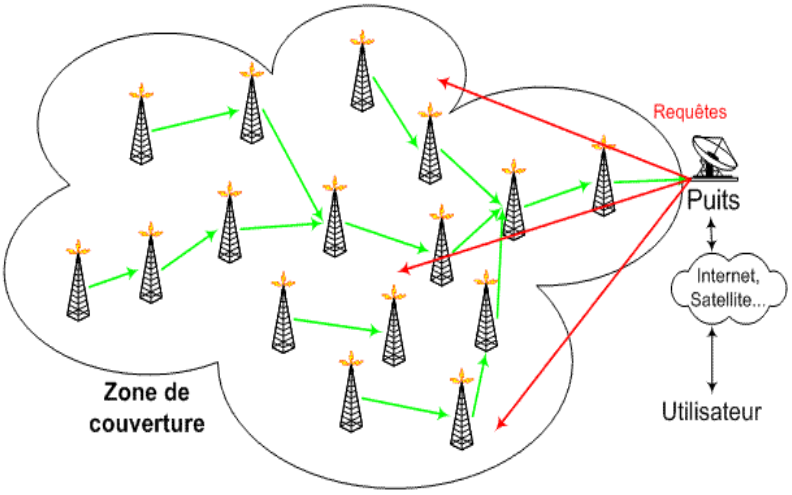


Figure I. 2 : communication a la demande [2]

3.5.2 Suite à un événement

Un événement se produit en un point de la zone de couverture (changement brusque de température, mouvement...), les capteurs situés à proximité remontent alors les informations relevées et les acheminent jusqu'au puits.[1]

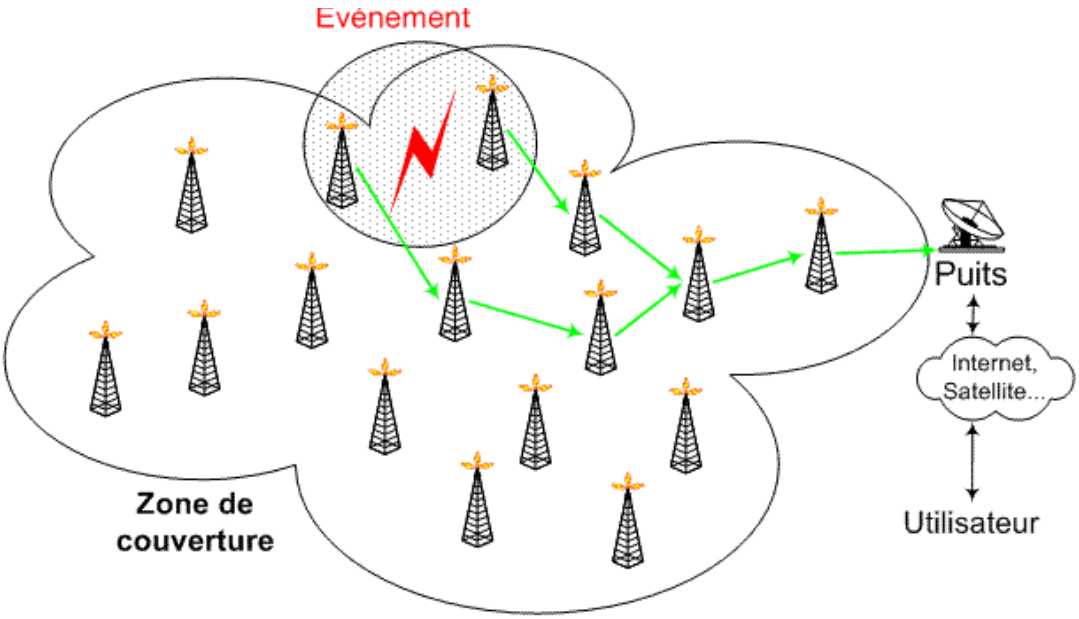


Figure I. 3 : communication suite à un événement [2]

3.6 Architecture protocolaire

Le rôle de ce modèle consiste à standardiser la communication entre les composants du réseau afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Ce modèle comprend 5 couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que 3 couches pour la gestion de la puissance d'énergie, la gestion de la mobilité ainsi que la gestion des tâches (interrogation du réseau de capteurs). Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction. Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur.[4]

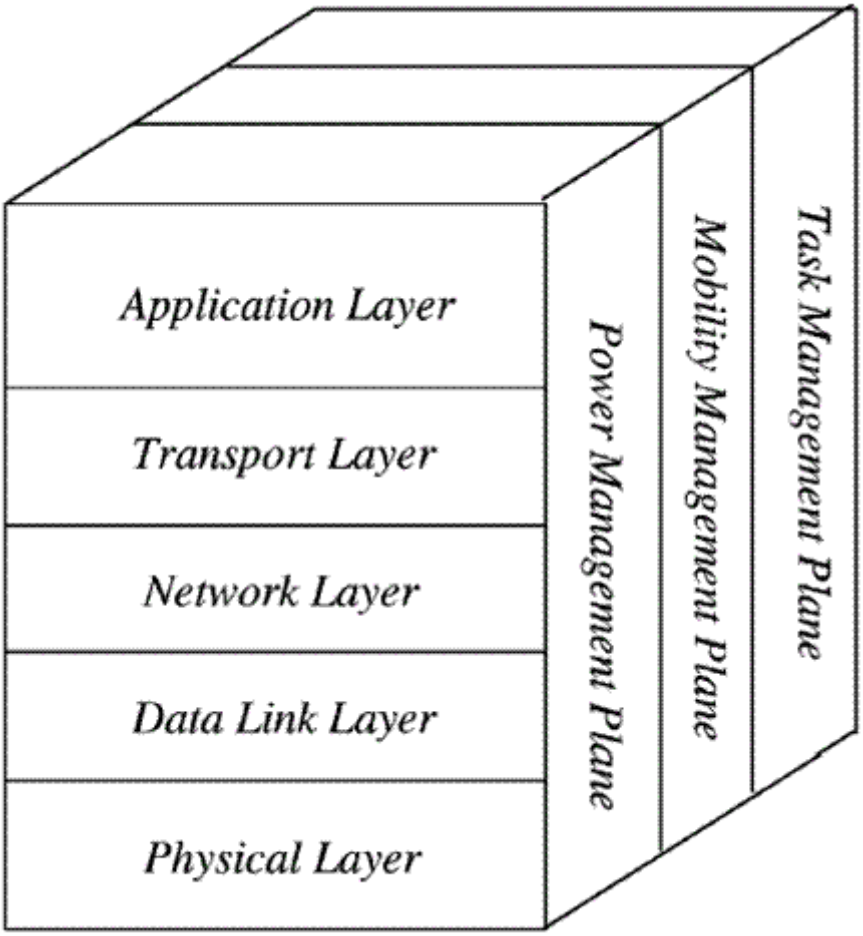


Figure I. 4 : Modèle en couches pour la communication dans les RCSF [5]

3.6.1 Rôle des couches :

La couche physique : Spécifications des caractéristiques matérielles, des fréquences porteuses, etc...

La couche liaison : Spécifie comment les données sont expédiées entre deux nœuds/routeurs dans une distance d'un saut. Elle est responsable du multiplexage des données, du contrôle d'erreurs, de l'accès au media,... Elle assure la liaison point à point et multipoint dans un réseau de communication.[4]

La couche réseau : Dans la couche réseau le but principal est de trouver une route et une transmission fiable des données, captées, des nœuds capteurs vers le puits "sink" en optimisant l'utilisation de l'énergie des capteurs. Ce routage diffère de celui des réseaux de transmission ad hoc sans fils par les caractéristiques suivantes:

il n'est pas possible d'établir un système d'adressage global pour le grand nombre de nœuds.

les applications des réseaux de capteurs exigent l'écoulement des données mesurées de sources multiples à un puits particulier.

les multiples capteurs peuvent produire de mêmes données à proximité d'un phénomène (redondance).

les nœuds capteur exigent ainsi une gestion soigneuse des ressources.

En raison de ces différences, plusieurs nouveaux algorithmes ont été proposés pour le problème de routage dans les réseaux de capteurs

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

La couche application : Cette couche assure l'interface avec les applications. Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels.[4]

3.6.2 Plans de gestion

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

Plan de gestion d'énergie : contrôle l'utilisation de la batterie. Par exemple, après la réception d'un message, le capteur éteint son récepteur afin d'éviter la duplication des messages déjà reçus. En outre, si le niveau d'énergie devient bas, le nœud diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage. L'énergie restante est réservée au captage .

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

Plan de gestion de tâche : balance et ordonnance les différentes tâches de captage de données dans une région spécifique. Il n'est pas nécessaire que tous les nœuds de cette région effectuent la tâche de captage au même temps ; certains nœuds exécutent cette tâche plus que d'autres selon leur niveau de batterie.[4]

3.7 Applications des RCSF

Les RCSF peuvent avoir beaucoup d'applications (voir figure suivantes). Parmi elles, nous citons :

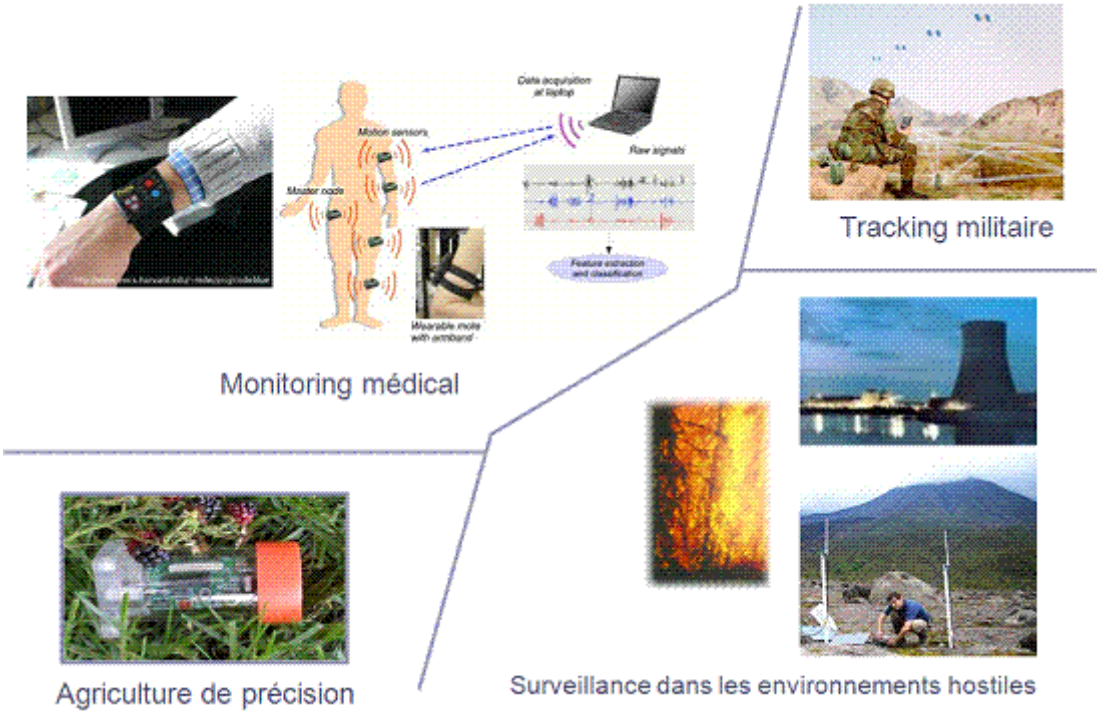


Figure I. 5 : Applications des RCSF [6]

-Découvertes de catastrophes naturelles : On peut créer un réseau autonome en dispersant les nœuds dans la nature. Des capteurs peuvent ainsi signaler des événements tel que feux de forêts, tempêtes ou inondations. Ceci permet une intervention beaucoup plus rapide et efficace des secours.

-Détection d'intrusions : En plaçant, à différents points stratégiques, des capteurs, on peut ainsi prévenir des cambriolages ou des passages de gibier sur une voie de chemin de fer (par exemple) sans avoir à recourir à de coûteux dispositifs de surveillance vidéo.

-Applications métier : On pourrait imaginer devoir stocker des denrées nécessitant un certain taux d'humidité et une certaine température (min ou max). Dans ces applications, le réseau doit pouvoir collecter ces différentes informations et alerter en temps réel si les seuils critiques sont dépassés. [7]

-Contrôle de la pollution : On pourrait disperser des capteurs au-dessus d'un emplacement industriel pour détecter et contrôler des fuites de gaz ou de produits chimiques. Ces applications permettraient de donner l'alerte en un temps record et de pouvoir suivre l'évolution de la catastrophe.

-Agriculture : Des nœuds peuvent être incorporés dans la terre. On peut ensuite questionner le réseau de capteurs sur l'état du champ (déterminer par exemple les secteurs les plus secs afin de les arroser en priorité). On peut aussi imaginer équiper des troupeaux de bétail de capteurs pour connaître en tout temps, leur position ce qui éviterait aux éleveurs d'avoir recours à des chiens de berger.

-Surveillance médicale : En implantant sous la peau de mini capteurs vidéo, on peut recevoir des images en temps réel d'une partie du corps sans aucune chirurgie pendant environ 24h. On peut ainsi surveiller la progression d'une maladie ou la reconstruction d'un muscle.

-Contrôle d'édifices : On peut inclure sur les parois des barrages des capteurs qui permettent de calculer en temps réel la pression exercée. Il est donc possible de réguler le niveau d'eau si les limites sont atteintes. On peut aussi imaginer inclure des capteurs entre les sacs de sables formant une digue de fortune. La détection rapide d'infiltration d'eau peut servir à renforcer le barrage en conséquence. Cette technique peut aussi être utilisée pour d'autres constructions tels que ponts, voies de chemins de fer, routes de montagnes, bâtiments et autres ouvrages d'art.[7]

4 Les systèmes d'exploitation pour les réseaux de capteurs

4.1 TinyOS :

TinyOS est un système d'exploitation open source conçu pour les réseaux de capteurs sans fil. Il est basé sur une architecture orientée composants qui favorise l'implémentation et l'innovation rapide. De plus, il génère un noyau de petite taille (quelques ko), comme l'exigent les contraintes de mémoire imposées par les réseaux de capteurs.[8]

La première implémentation de TinyOS a été réalisée a l'université de Berkeley en 1999 [24]. La version 1.0 est sortie en septembre 2002. La version la plus récente de ce système est la 2.0.2 qui a été disponible en juillet 2007.

Ce système d'exploitation est développe avec le langage NesC. Le développement et la maintenance de ce système est maintenant sous la responsabilité d'un consortium international, TinyOS Alliance.[8]

4.2 Contiki :

Contiki est un système d'exploitation portable, open source et multitâche pour les réseaux de capteurs. Il supporte beaucoup de plateformes et il a un environnement de simulation netsim.[9]

Ce système a été développe par l'équipe des systèmes embarques dans l'institut des sciences informatique suédois. Contiki supporte le multitâche et il implémente la pile protocolaire TCP/IP. Il ne consomme pas beaucoup de mémoire : quelques ko de code et quelques centaines d'octets dans la RAM.[10]

Au contraire de TinyOS qui se base sur la notion d'événements, celui-ci se base sur le multitâche et l'édition statique des liens.

5 Conclusion

Dans ce chapitre, nous avons rappelé certains concepts de base du réseau de capteurs sans fil, du matériel et des logiciels des nœuds de capteurs. En outre, nous avons décrit leurs différentes architectures et caractéristiques. Enfin, les nombreuses applications de la vie réelle de RCSF sont présentées. Dans le chapitre suivant, nous passerons en revue les systèmes de détection d'intrusion, le champ d'application de notre étude.

CHAPITRE 02

Sécurité et systèmes de détection d'intrusion IDS dans RCSF

1 Introduction

Les réseaux de capteurs sans fil peuvent s'apparenter aux réseaux ad hoc sans fil, mais ils se différencient par plusieurs limitations, ce qui ne permet pas d'appliquer directement les solutions de sécurité existantes. Pour pouvoir développer des solutions de sécurité adaptées à ces réseaux, il faut bien comprendre leurs contraintes.

Dans ce chapitre, on parle sur les conditions de sécurité et on détaillera les obstacles de sécurité liés aux réseaux de capteurs sans fil. Ensuite, nous décrirons l'IDS et ses moteurs d'analyse qui lui permettent de distinguer les différents comportements.

Nous aborderons également les techniques de détection d’intrusion. En fin on énumérera les attaques possibles dans ces réseaux.

2 le but De Sécurité

Un réseau de capteur est un type spécial de réseaux. Il partage quelques vulgarisations avec un réseau informatique typique, mais pose également des conditions uniques de ses propres caractéristiques. Par conséquent, un protocole de sécurité pour un RCSF, doit satisfaire une ou plusieurs conditions de sécurité [8], à savoir :

2.1 Confidentialité Des Données :

La confidentialité des données est la question la plus importante dans la sécurité de réseau. L'approche standard pour sécuriser le transfert des données est de crypter les données avec une clef secrète connue par l’émetteur et le récepteur [8].

2.2 Intégrité des données :

Un nœud intrus (adversaire) peut modifier les données transférées. Par exemple, un nœud malveillant peut ajouter quelques fragments ou manœuvrer les données dans un paquet.

Ce nouveau paquet peut alors être envoyé au récepteur original. La perte ou les dommages de données peut même se produire sans présence d'un nœud malveillant dû à l'environnement dur de communication. Ainsi, l'intégrité des données s'assure qu'aucune donnée reçue n'a été changée en transit [11].

2.3 Fraîcheur De Données :

Même si la confidentialité et l'intégrité des données sont assurées, nous devons également assurer la fraîcheur de chaque message. Officieusement, la fraîcheur de données suggère que les données soient récentes, et elles s'assurent qu'aucun vieux message n'a été rejoué. Cette condition est particulièrement importante quand il y a des stratégies de partager clef utilisées dans la conception. Des clefs typiquement partagées doivent être changées avec le temps.

Cependant, cela prend du temps pour de nouvelles clefs partagées d'être propagées au réseau entier. Dans ce cas-ci, il est facile pour l'adversaire d'employer une attaque de rejouer.

Pour résoudre ce problème un compteur relatif au temps différent, peut être ajouté dans le paquet pour assurer la fraîcheur de données [11].

2.4 Auto-Organisation :

Un réseau de capteur sans fil est typiquement un réseau adhoc, qui exige chaque nœud capteur soit indépendant et assez flexible à l'auto organisation. Il n'y a aucune infrastructure fixe disponible pour la gestion de réseau dans un réseau de capteurs. L'auto organisation apporte un grand défi à la sécurité du réseau de capteurs sans fil [11].

2.5 La Localisation :

Souvent, l'utilité d'un réseau de capteur se fondera sur ses capacités de localiser automatiquement chaque capteur dans le réseau. Un réseau de capteurs conçu pour détecter des anomalies aura besoin de l'information précise d'endroit afin d'indiquer exactement l'endroit d'un défaut [11].

2.6 Authentification :

Un adversaire n'est pas limité simplement à modifier le paquet de données. Il peut changer le jet entier de paquets en injectant les paquets additionnels. Ainsi le récepteur doit s'assurer que les données utilisées dans n'importe quel processus décisionnel proviennent de la source correcte.

D'autre part, en construisant le réseau de capteurs, l'authentification est nécessaire pour beaucoup de tâches administratives (coefficient de reprogrammation ou de contrôle). D'après ce qui précède, nous pouvons voir que l'authentification de message est importante pour beaucoup d'applications dans les réseaux de capteurs. Officieusement, l'authentification de données permet à un récepteur de vérifier que les données sont vraiment envoyées par l'expéditeur réclamé.

Dans le cas de communication bipartite, l'authentification de données peut être réalisée par un mécanisme purement symétrique : l'expéditeur et le récepteur partagent une clef secrète pour calculer le code d'authentification de message (IMPER) de toutes les données communiquées [11] [12].

3 Les obstacles de sécurité liés aux réseaux de capteurs

3.1 Des ressources limitées :

L'utilisation des algorithmes de sécurité nécessitent des ressources de mémoires pour la mémorisation du code et des données, des ressources en énergie et en calcul.

Plateforme	microcontrôleur	RAM	Memoire programme	Puce radio
Mica2	ATMega128 (7,3728MHz)	4 ko	128 ko	CC1000 (868-916MHz)
Tmote Sky	MSP430 (8MHz)	10 ko	48 ko	CC2420 (2400-2483MHz)

Tableau II. 1: Comparaison des caractéristiques des deux plateformes Tmote Sky et Mica2[13]

- mémoire et espace de stockage limites : Le capteur est un composant miniature avec un espace mémoire et de stockage limite, et avec une faible vitesse de calcul.

  Dans le tableau 2.1, on a une comparaison entre les deux capteurs les plus connus.

  Le capteur Mica2 a un processeur de fréquence 7,3728 MHz et une mémoire pour le programme de 128 ko. Sur le même capteur, on doit installer le code du système d'exploitation et de l'application. Donc le code de sécurité et les données relatives doivent être très petits.

- Limitation en énergie : C'est la contrainte la plus forte, puisque généralement les capteurs sont déployés a des endroits inaccessibles ou d'accès difficile, donc on ne peut pas changer les batteries ou les recharger (c'est cher a utiliser dans un capteur). Alors la ressource d'énergie embarquée avec les capteurs doit être conservée pour étendre leur vie et par la suite celle du réseau entier. L'ajout des fonctions de sécurité a un effet a prendre en considération sur la consommation des ressources, de temps processeur et par la suite sur la consommation en énergie [11].

Le tableau 3.2 présente le nombre de cycles CPU et le temps nécessaire pour chiffrer un message de longueur 29 octets par différents algorithmes de chiffrement.

Le tableau 3.1 présente les mêmes paramètres mais pour la génération du MAC, tous les algorithmes opèrent en mode CBC-MAC. Ces tests sont faits sur la plateforme mica2.

Algorithme	Temps (ms)	Cycles CPU	Energie (j)
Skip Jack	2.16	15,925.2	51.84
RC5	1.50	11,059.2	36.00
RC6	10.78	79,478.7	258.72
TEA	2.56	18,874.4	61.44
DES	608.00	4, 482, 662,4	14,592.00

Tableau II. 2: L'impact du chiffrement de 29 octets sur la consommation CPU [14]

Algorithme	Temps (ms)	Cycles CPU	Energie (j)
Skip Jack	2.99	22,044.6	71.76
RC5	2.08	15,335.4	49.92
RC6	15.84	116,785.2	380.16
TEA	5.07	37,380.1	121.68
DES	1,208.00	8,906,342.4	28,992.00

Tableau II. 3: La consommation CPU lors du calcul du MAC d'un paquet de 29 octets[15]

3.2 Communication non fiable :

Cette caractéristique est héritée des réseaux sans fil. Les données sont transmises dans l'air, donc chaque capteur qui se trouve dans le rayon de couverture peut écouter les messages échangés. L'application d'un bruit sur le canal peut rendre les capteurs incapables de transmettre les messages vu que le media peut apparaître comme occupé en permanence [11].

3.3 Les risques inattendus :

Selon l'application du réseau de capteurs, les capteurs sont sans surveillance ou surveillées après une longue période. Les capteurs sont sans surveillance lorsqu'ils sont, par exemple, déployés derrière les lignes de l'ennemi. Les mises en garde aux capteurs sans surveillance sont :

- exposition aux attaques physiques : Le capteur est généralement déployé dans un environnement ouvert aux ennemis, et peut faire face à des conditions climatiques difficiles.
- gestion à distance : La gestion à distance du réseau rend la détection d'une attaque physique (compromission de capteur) et la maintenance des capteurs (rechange ou recharge de batterie) impossible.
- pas de station de base : Le réseau de capteurs doit être conçu pour être un réseau distribué sans point de gestion central. Mais s'il y a une erreur de conception, l'organisation du réseau peut devenir difficile, inefficace et fragile [10].

4 Les IDS

4.1 Définition d’un IDS :

Un IDS est une sonde placée judicieusement sur un réseau ou un système, et qui va repérer les activités douteuses ou anormales sur cette cible et alerter les responsables sécurité. De cette façon, on peut obtenir une connaissance des tentatives réussies (ou non) d'attaque ou d'intrusion sur le système. On différencie plusieurs types d'IDS, à savoir les NIDS (ou Network Intrusion Détection System), qui se basent sur des analyses réseau, les HIDS (ou Host Intrusion Détection System), qui surveillent l'activité d'un hôte, et enfin les IDS hybrides, qui combinent HIDS et NIDS [16].

- Les NIDS Ces outils analysent le trafic réseau; ils comportent généralement une sonde qui "écoute" sur le segment de réseau à surveiller et un moteur qui réalise l'analyse du trafic afin de détecter les signatures d'attaques ou les divergences face au modèle de référence. Les IDS Réseaux à base de signatures sont confrontés actuellement à deux problèmes majeurs qui sont : l'utilisation grandissante du cryptage, et des réseaux commutés. En effet, il est d'une part plus difficile " d'écouter " sur les réseaux commutés et le cryptage rend l'analyse du contenu des paquets presque impossible. La plupart des NIDS sont aussi dits IDS inline car ils analysent le flux en temps réel. Pour cette raison, la question des performances est très importante. De tels IDS doivent être de plus en plus performants afin d'analyser les volumes de données de plus en plus importants pouvant transiter sur les réseaux [17].

Figure II. 1 : Placement d’un NIDS en amont d’un pare-feu .

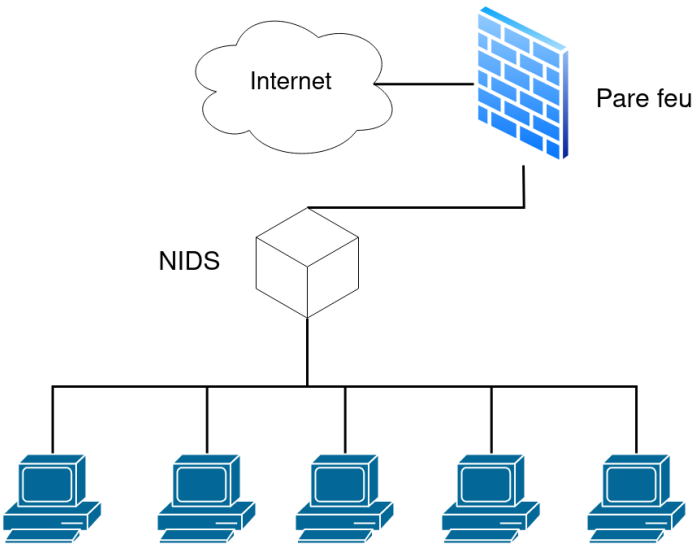


Figure II. 2 : Placement d'un NIDS en aval d'un pare-feu.[18]

- Les HIDS Les IDS Systèmes analysent quant à eux le fonctionnement ou l'état des machines sur lesquelles ils sont installés afin de détecter les attaques. Pour cela ils auront pour mission d'analyser les journaux systèmes, de contrôler l'accès aux appels systèmes, de vérifier l'intégrité des systèmes de fichiers ... Ils sont très dépendants du système sur lequel ils sont installés. Il faut donc des outils spécifiques en fonction des systèmes déployés. Ces IDS peuvent s'appuyer sur des fonctionnalités d'audit propres ou non au système d'exploitation, pour en vérifier l'intégrité, et générer des alertes. Il faut cependant noter qu'ils sont incapables de détecter les attaques exploitant les faiblesses de la pile IP du système, typiquement les Dénis de service comme SYN FLOOD ou autre [17].

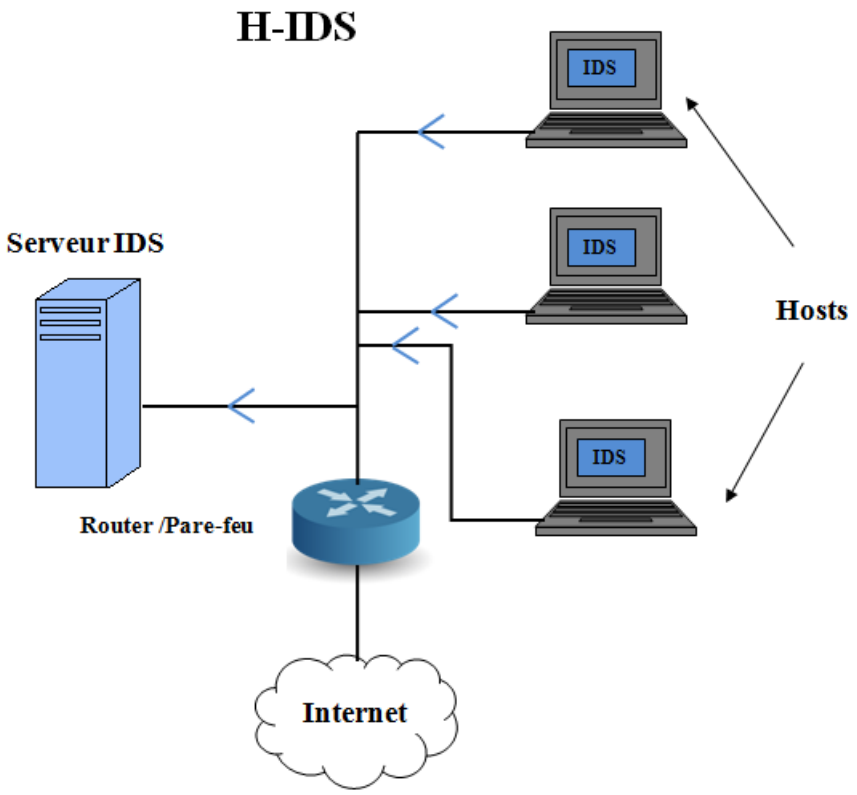


Figure II. 3 : les IDS HIDS.[19]

- Les IDS hybrides (NIDS+HIDS) Les IDS hybrides rassemblent les caractéristiques de plusieurs IDS différents. En pratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil, de surveiller le réseau et les terminaux. Les sondes sont placées en des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout, et agréger/lier les informations d'origines multiples [17].

4.2 Les différents types d'intrusion :

1) Le cheval de Troie :

Programme malveillant d'apparence anodine (jeux, utilitaire) qui, une fois installé, peut causer des dégâts ou permettre la prise de contrôle à distance de l'ordinateur

infecté, et ainsi fournir au pirate, une porte ouverte au contenu du système d'information de l'entreprise [20].

2) Le ver (Worm) :

Programme capable de fonctionner de manière indépendante. Il se propage de machine en machine. Un ver ne modifie aucun programme, mais il peut transporter des séries de codes informatiques qui pourront le faire (des virus par exemple) [14].

3) Le virus :

Capable d'infecter d'autres programmes en les modifiant pour y inclure une copie de lui même. Le virus nécessite l'exécution du programme hôte pour s'activer. Il se multiplie au sein de l'environnement qu'il cible et entraine corruption, perturbation et/ou destruction [20].

4) Les bombes logiques :

Programmes destructeurs à déclenchement différé. Par exemple, un programmeur insère dans le programme de paie de l'entreprise qui l'emploie une fonction de destruction dont l'exécution sera déclenchée si le nom du programmeur disparaît du fichier du personnel [20].

5) L’attaque en déni de service :

Activité consistant à empêcher quelqu'un d'utiliser un service par saturation d'exécution de programmes.

6) Le pourriel ou spam :

Communication électronique, non sollicitée par les destinataires et expédiée en masse à des fins publicitaires ou malhonnêtes [20].

7) L’adware :

Permet d'afficher des bannières publicitaires [20].

8) Le spyware :

Installe sur le poste de l'utilisateur un logiciel espion et envoient régulièrement et sans accord préalable de l’utilisateur, des informations statistiques sur les habitudes de celui-ci [20].

9) Le hameçonnage :

consiste à adresser à une personne un courriel de sa banque par exemple, lui expliquant qu'à la suite d'un problème informatique ou d'un changement de logiciel, cette personne doit confirmer ses codes d'accès pour pouvoir continuer à consulter ses comptes, faire des transferts de fond etc. Dans ce message figure un lien permettant d'accéder à la page de confirmation des codes en question. Evidemment cette page web reproduit fidèlement la charte graphique du site officiel de l'institution bancaire, ce qui ne manquera pas de rassurer l'utilisateur alors convaincu de dialoguer avec sa banque [20].

4.3 Méthodes de détection d’intrusion

Les systèmes de détection d’intrusion se différencient en matière de méthodes utilisées pour détecter les intrusions. Initialement, il existait deux grandes familles de techniques de détection, mais avec l’arrivée des IDS distribués (DIDS), de nombreuses techniques ont vu le jour. Nous détaillons dans cette partie les méthodes de détection utilisées par ces systèmes.

4.3.1 La détection d’anomalies :

La détection d'anomalie de comportement est une technique assez ancienne elle est utilisée également pour détecter des comportements suspects en téléphonie, Cette technique basée sur le comportement « normal » du système.

- ❖ Une déviation par rapport à ce comportement est considérée suspecte.
- ❖ Le comportement doit être modélisé on définit alors un profil.
- ❖ Une attaque peut être détectée sans être préalablement connue.

Avantages

1. permet de détecter de nouvelles attaques sans qu'il y ait besoin de redéfinir la politique de sécurité.
2. Facilite la création de règles adaptées à ces attaques.
3. Difficile à tromper.

Inconvénients

1. Le point noir de cette approche est le grand nombre de fausses alarmes dues aux comportements imprévisibles des utilisateurs du réseau.
2. Générer un profil est complexe
3. Diagnostics long et précis en cas d'alerte
4. Elle exige souvent l'historique à long terme des évènements enregistrés afin de caractériser les modèles normaux de comportement. Les systèmes basés sur cette approche doivent être dotés d'une certaine intelligence pour raison d'apprentissage automatique.

4.3.2 La reconnaissance de signature :

Cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Ce type d'IDS est purement réactif ; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite des mises à jour fréquentes.

Avantages

1. Très efficace pour détecter des attaques sans produire un grand nombre de fausses alarmes.
2. Peut rapidement et sûrement diagnostiquer l'utilisation d'un outil spécifique ou une technique d'attaque.
3. Ceci peut aider les responsables de sécurité à donner la priorité aux mesures correctives.

Inconvénients

1. Peut seulement détecter les attaques connues, dont les signatures sont introduites dans le système, donc le système de détection doit être constamment mis à jour avec les signatures des nouvelles attaques.
2. Beaucoup de systèmes adoptant cette approche sont conçus pour employer un nombre limité de signatures qui peuvent être définis, ce qui les empêchent de détecter des variantes de ces attaques.

Une fois une attaque détectée, un IDS a le choix entre plusieurs types de réponses, que nous allons maintenant détailler.

4.4 Les types de réponse:

Il existe deux types de réponses, suivant les IDS utilisés. La réponse passive est disponible pour tous les IDS, la réponse active est plus ou moins implémentée [21,22,23].

4.4.1 Réponse active :

La réponse active au contraire a pour but de stopper une attaque au moment de sa détection [21,22,23].

4.4.2 Réponse passive:

La réponse passive d’un IDS consiste à enregistrer les intrusions détectées dans un fichier de log qui sera analysé par le responsable de sécurité. Certains IDS permettent de d’enregistrer l’ensemble d’une connexion identifiée comme malveillante. Ceci permet de remédier aux failles de sécurité pour empêcher les attaques enregistrées de se reproduire, mais elle n’empêche pas directement une attaque de se produire [21,22,23].

5 Les attaques dans les réseaux de capteurs

Les attaques de sécurité possibles dans les réseaux de capteurs sont:

5.1 Écoute passive du réseau :

L'attaquant, qui dispose d'un équipement puissant (vitesse de calcul, espace de stockage, ressource en énergie, etc.), collecte les informations échangées dans le réseau de capteurs si elles ne sont pas chiffrées.

5.2 Compromission du nœud :

En compromettant un nœud, l'attaquant peut récupérer les informations incluses : Programme, clés cryptographiques [24].

5.3 Injection de nœuds malveillants :

L'attaquant peut ajouter dans le réseau des nœuds malveillants pour injecter des données falsifiées dans le réseau. D'habitude ces nœuds seront plus robuste pour attirer les messages des autres nœuds, car dans les réseaux de capteurs sans fil, le choix du meilleur chemin par les algorithmes de routage se base sur plusieurs paramètres ; parmi lesquelles l'énergie et la puissance de calcul des nœuds.

5.4 Le mauvais fonctionnement d'un nœud :

Le mauvais fonctionnement d'un nœud peut générer des données inexactes qui peuvent mettre en péril l'intégrité du réseau, et surtout quand ce nœud joue un rôle important dans l'agrégation des données. Par exemple un chef de la grappe (cluster).

5.5 La panne d'un nœud :

La panne d'un nœud peut affecter le fonctionnement du réseau, surtout quand ce nœud est un chef de grappe. Alors le protocole de routage doit être robuste pour annuler l'effet d'une telle panne en construisant des chemins de routage alternatifs.

5.6 La corruption de message :

Quand le contenu d'un message est modifié par un attaquant, il compromet l'intégrité du message.

5.7 L'analyse du trafic :

Cette analyse peut se faire même si les messages sont chiffrés. En analysant les modes de communications et les activités des nœuds, l'attaquant peut déduire des informations sur l'organisation du réseau, par exemple sur les chefs de grappes.

5.8 Les boucles de routage :

Ce type d'attaque vise les informations échangées entre les nœuds. Le re-jeu ou la modification des informations de routage par un attaquant génère de faux messages d'erreurs. Les boucles de routage attirent ou repoussent le trafic dans le réseau.

5.9 Transmission sélective :

Supposons que tous les nœuds participent a la propagation des messages dans le réseau. Dans cette attaque, le nœud malveillant supprime quelques messages au lieu de les transmettre.

L'efficacité de cette attaque dépend de deux facteurs. Le premier est la place du nœud malveillant, plus il est proche de la station de base plus il recroit de messages. Deuxièmement est le pourcentage des messages qu'il supprime.

5.10 Trou noir (sinkhole) :

Le nœud malveillant se place à un endroit stratégique (proche de la station de base par exemple) et supprime tous les messages qu'il doit retransmettre. Cette attaque est grave lorsqu'il n'y a qu'une seule station de base dans le réseau.

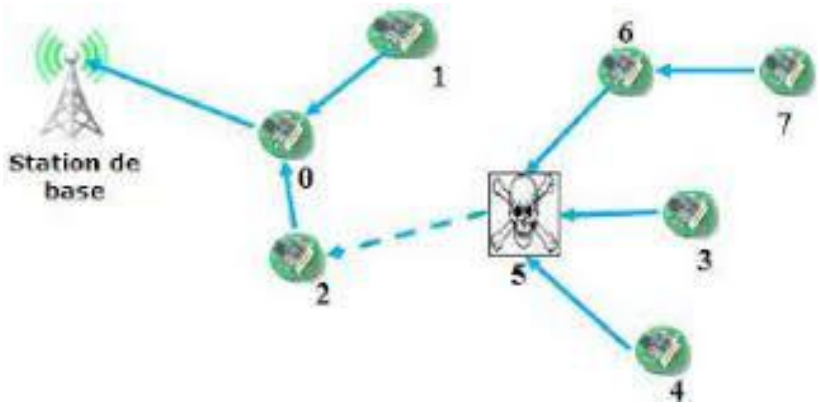


Figure II. 4 : exemple de Trou noir [25]

5.11 Usurpation d'identités (Sybil attacks) :

Dans ce type d'attaques, le nœud malveillant prend un grand nombre d'identités qui peuvent être volés ou imaginaires. Cette attaque peut être utilisée contre les protocoles de routage.

5.12 Réplication de nœud (clonage) :

L'attaquant récupère un nœud. Il en extrait les secrets et il les transfère à des nœuds génériques. En fin, il déploie ces nœuds. Alors il peut injecter de fausses données et supprimer des données légitimes [26].

5.13 Trou de ver (wormhole) :

L'intrus capture un message et, en utilisant un canal de faible latence, le retransmet vers un lieu distant dans le réseau. Le canal ainsi crée fait transiter un message a un endroit du RCSF auquel il ne devrait normalement pas arriver, ou sinon avec une plus grande latence. Cette attaque a une influence notable sur le routage dans le réseau.

5.14 Attaque par inondation avec le message HELLO :

Le nœud malveillant diffuse un message Hello dans le réseau avec une grande énergie d'émission, tout en prétendant qu'il provient de la station de base. Les receveurs de ce message essayeront de transmettre tous leurs messages a travers le nœud malveillant, car ils pensent qu'il appartient au plus proche chemin vers la station de base. Ce qui gaspillera l'énergie des nœuds en transmettant des messages inutilement.

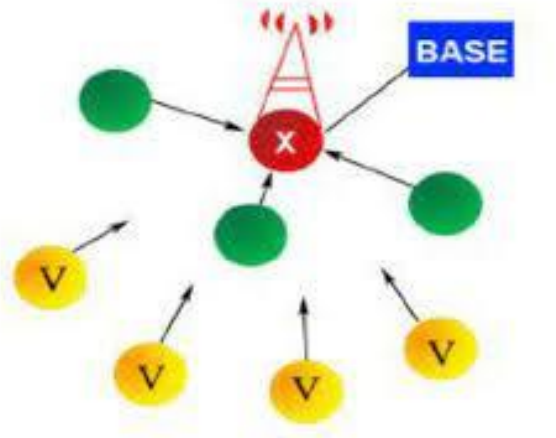


Figure II. 5 : attaque hello flooding [27]

5.15 Les attaques de déni de service par interférence :

Le nœud malveillant inonde avec du bruit les fréquences radio utilisées par le réseau de manière a empêché les transmissions et/ou les réceptions de messages. Ce type d'attaques peut affecter tout ou partie du réseau selon la portée radio de l'intrus. Dans ce cas-là, l'intention est de provoquer un déni de service [28].

5.16 Brouillage radio :

Un attaquant va envoyer des ondes sur la même fréquence que le réseau de capteurs sans fil [WS02]. Ainsi les nœuds ne pourront plus communiquer car le médium est saturé par le brouillage radio.

5.17 Insertions de boucles infinies :

Un attaquant va modifier le routage du réseau avec un ou plusieurs nœuds malicieux, dans le but d’envoyer des messages qui vont être routés en boucles infinies et vont donc consommer l’énergie du réseau.

5.18 Ralentissement :

Un attaquant peut programmer des nœuds malicieux qui seront comme des agents dormant et qui n’auront que pour but de ralentir l’information (par exemple avec une attaque de type trou gris.

6 Conclusion

On a parler dans ce chapitre sur des généralités de la sécurité dans le réseau de capteur sans fil et des systèmes de détection d’intrusion (les IDS) et on définit quelques attaques qui touche ce type de réseau parmi les, attaque par déni de service (DOS).

Notre travaille c’est que proposer un nouveau système qui détecter ce type des attaques (DOS), dans le chapitre 03 il ya explication sur notre IDS avec une implémentation bien détaillé.

CHAPITRE 03

Notre approche : IDS basé sur MANOVA

1 Introduction

Il existe de nombreux attaques qui menacé les réseaux capteurs sans fils , le plus important étant les attaques DOS, Ce qui a fait de nombreux chercheurs à trouver des solutions pour résoudre ces problèmes en construisant des systèmes de détection des intrusions(IDS).

Pour résoudre ce problème, nous avons proposée un IDS basé sur l'analyse de variance multi variée (MANOVA).

Dans ce chapitre, nous expliquons comment fonctionne notre IDS, et pour ce faire nous allons concevoir un ensemble de données et notre système analyser les attaques dos en fonction d'un ensemble des attributs, puis nous décrirons notre étude expérimentale et ses résultats présentés.

2 Schémas de détection proposé

La méthode de détection est basée sur MANOVA

2.1 Modèle MANOVA :

L'analyse de variance multi variée ou MANOVA pour « Multivariate analysis of variance » est une méthode d'analyse similaire à ANOVA.
Contrairement à ANOVA, MANOVA utilise la variance-covariance entre les variables aléatoires lors du test de la signification statistique des différences de moyennes..[29]

Manova teste les effets de facteurs sur plusieurs variables réponses, il est donc possible de tester conjointement toutes les hypothèses testées par une série d'ANOVA avec plus de chance d'observer un effet significatif.

Dans le processus de vérification de la significativité des différences entre paramètres correspondants à différentes modalités d'un facteur, de nombreux tests ont été proposés à cet escient :

- Test de Wilks Lambda
- Test de la trace de Hotelling- Lawley
- Test de la trace de Pillai

- Test de la plus grande racine de Roy.[30]

Dans MANOVA , nous avons *m* vecteurs aléatoires X_1 ,…, X_m (représentant des groupes ou des traitements). Chaque X_j est un vecteur colonne $k \times 1$ de forme

$$\begin{bmatrix} x_{j1} \\ \cdots \\ x_{jk} \end{bmatrix} \quad (1)$$

où chaque x_{jp} est une variable aléatoire.

Pour chaque vecteur aléatoire X_j nous collectons un échantillon { X_{1j},…, $X_{n_j j}$ } de taille n_j . Nous définissons également $n = \sum_{j=1}^{m} n_j$. Chaque échantillon X_{ij} est un vecteur $k \times 1$ de forme

$$\begin{bmatrix} x_{ij1} \\ \cdots \\ x_{ijk} \end{bmatrix} \quad (2)$$

où chaque x_{ijp} est un élément de données (pas une variable aléatoire), où l'indice *i* fait référence au sujet de l'expérience ($1 \le i \le n_j$), l'indice *j* fait référence au groupe ($1 \le j \le m$) et l'indice *p* fait référence à la position (c.-à-d. variable dépendante) dans le vecteur aléatoire ($1 \le p \le k$).

Notre objectif est de tester l' hypothèse nulle

$H_0 : \mu_1 = \mu_2 = \cdots = \mu_m$ où les μ_j sont des vecteurs

$$\begin{bmatrix} \mu_{j1} \\ \cdots \\ \mu_{jk} \end{bmatrix} \quad (3)$$

Nous définissons maintenant les différents moyens.

$$\bar{X}_T = \begin{bmatrix} \bar{x}_1 \\ \cdots \\ \bar{x}_k \end{bmatrix} \quad (4)$$

Où

$$\bar{x}_p = \frac{1}{n}\sum_{j=1}^{m}\sum_{i=1}^{n_j} x_{ijp} \quad (5)$$

Le vecteur moyen du groupe échantillon pour le groupe *j* est un vecteur colonne

$$\bar{X}_j = \begin{bmatrix} \bar{x}_{j1} \\ \cdots \\ \bar{x}_{jk} \end{bmatrix} \quad (6)$$

où

$$\bar{x}_{jp} = \frac{1}{n_j}\sum_{i=1}^{n_j} x_{ijp} \quad (7)$$

Le total est :

$$T = \sum_{j=1}^{m}\sum_{i=1}^{n_j}(X_{ij} - \bar{X}_T)(X_{ij} - \bar{X}_T)^T \quad (8)$$

Nous définissons l' hypothèse **H** comme **:**

$$H = \sum_{j=1}^{m} n_j(\bar{X}_j - \bar{X}_T)(\bar{X}_j - \bar{X}_T)^T \quad (9)$$

Et la somme des erreurs **E** comme **:**

$$E = \sum_{j=1}^{m}\sum_{i=1}^{n_j}(X_{ij} - \bar{X}_j)(X_{ij} - \bar{X}_j)^T \quad (10)$$

Propriété : $T = H + E$

On utilise quatre statistiques de test différentes basées sur la table MANOVA :
Le test de wilk le plus utilisé.[31]

Lambda de Wilk: $\Lambda = \frac{|E|}{|H+E|}$

nous rejetons l'hypothèse nulle lorsque le Lambda de Wilk est proche de zéro.

P value : En statistique, la p-valeur est la probabilité d'obtenir les résultats observés d'un test, en supposant que l'hypothèse nulle est correcte. C'est le niveau de signification marginale dans un test d'hypothèse statistique représentant la probabilité de l'occurrence d'un événement donné.

Dans MANOVA :

Si P< 0.05 il existe un déférence entre les moyennes des groupes.

Si P > 0.05 il n'existe aucune déférence entre les moyennes des groupes.

2.2 Comparaison entre manova et anova

Les différences peuvent être résumées dans le tableau suivant :

ANOVA	MANOVA
-ANOVA vérifie les différences entre les moyennes de deux échantillons / populations. **-ANOVA concerne un variable,.** **-ANOVA utilise F test pour significativité test**	- MANOVA vérifie les différences entre plusieurs échantillons / populations.. - MANOVA concerne les différences dans plusieurs variables simultanément. -MANOVA utilise la relation covariance-variance. -MANOVA utilise : - Test de Wilks Lambda - Test de la trace de Hotelling- Lawley - Test de la trace de Pillai - Test de la plus grande racine de Roy.

Tableau III. 1:commparaison entre ANOVA et MANOVA

2.3 Méthode de détection :

L'attaque DoS vise à perturber le fonctionnement normal et les informations (attributs) du réseau. Avant de mettre en place notre méthode de détection, il nous a fallu déterminer quels sont les attributs les plus sensibles aux attaques DoS, pour les utiliser comme des indices révélateurs d'attaques DoS. Ensuite le Sink, peut effectuer le test afin de déterminer la présence d'une attaque. Notre algorithme suit les étapes suivantes :

- Le sink collecte les valeurs des attributs choisis au préalable pour une période de temps fixe, ces valeurs seront la référence d'un fonctionnement normal du réseau.
- Après, le sink collecte à nouveau des nouvelles valeurs qui seront comparais en utilisant MANOVA avec les valeurs références.
- Si le test échoue, on considère qu'il n'y a pas d'attaque sur le réseau et le sink passe à des nouvelles valeurs.
- Sinon, le sink alerte le réseau sur la présence d'une attaque DoS.

Dans la partie suivante nous détaillons comment nous avons implémenté notre approche, les outils qu'on a utilisés ainsi que les expérimentations qu'on a effectuées.

3 Simulations et Expérimentations

Pour tester notre approche, nous avons fait appel à la simulation pour générer des données de tests. Pour vérifier l'efficacité de notre approche. Dans la suite, nous détaillons comment nous avons généré ces données à partir de la simulation

3.1 Logiciels et outils utilisés :

a)Oracle VM Virtual Box : Oracle VM Virtual Box est une application multiplateforme gratuite, open source pour la création, la gestion et l'exécution de machines virtuelles (VM). Les machines virtuelles sont des ordinateurs dont les composants matériels sont émulés par l'ordinateur hôte.

b)NS2 : est un outil logiciel de simulation de réseaux informatiques, il est parmi les simulateurs les plus utilisés dans les laboratoires de recherche, afin de simuler et étudier les performances des protocoles réseau. Il offre une plateforme de développement de nouveaux protocoles et permet de les tester.[32]

c)Gawk : Gnu awk est un parseur de fichiers textes très simples à utiliser. Ils permettent de manipuler efficacement des fichiers textes de données délimitées par un caractère. Avec une syntaxe très facile à appréhender, les opérations pour filtrer des lignes, filtrer des colonnes, enrichir le contenu, convertir des formats, calculer des agrégats (moyennes, sommes par exemple), etc…[33]

d)*R Studio* : est un environnement de développement gratuit, libre et multiplateforme pour R, un langage de programmation utilisé pour le traitement de données et l'analyse statistique.

e) SPSS : SPSS est un logiciel spécialement conçu pour les analyses statistiques en sciences sociales. Il signifie Statistical Package for Social Sciences.[34]

3.2 Vue globale :

On a suivez plusieurs étapes afin de construire un système IDS, ce travail est illustré dans la figure III.1.

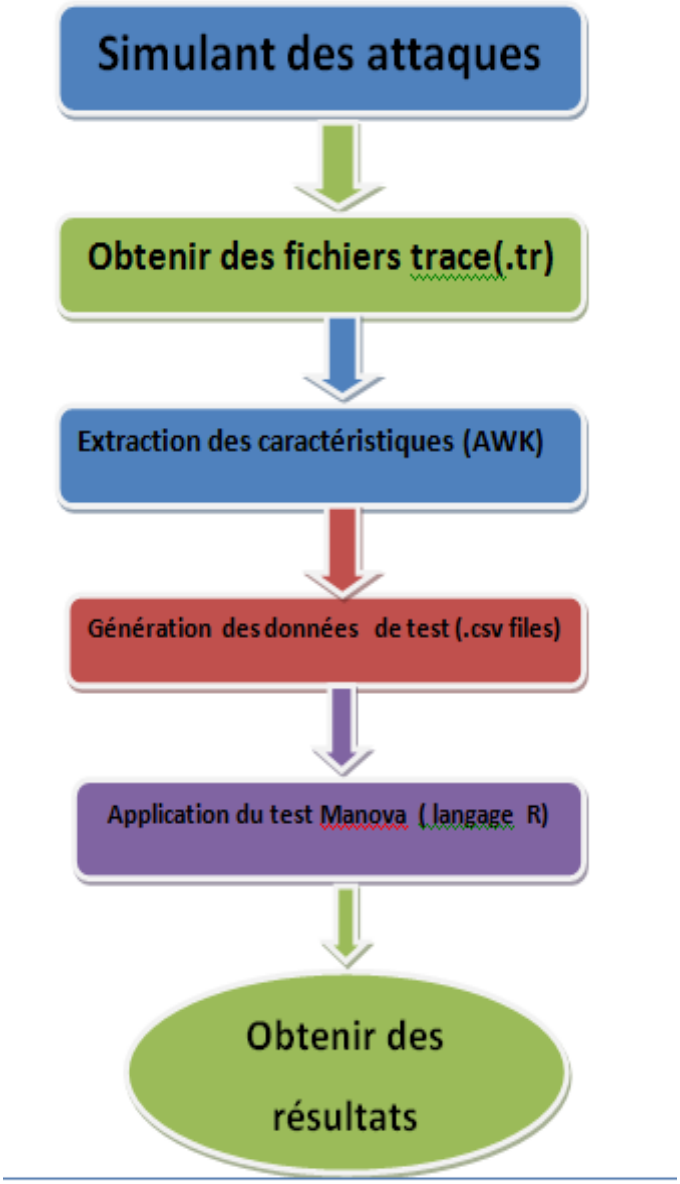


Figure III. 1:Vue globale du système

3.3 Simulation avec ns2 :

3.3.1 Modèle d'application :

Figure III. 2: Modèle d'application.

Nous avons considéré un environnement simple, où les nœuds de capteurs sont statiques et placés dans une topologie maillée comme le montre la Figure III.2. Le nœud avec l'identification "0" est considéré comme le Sink(puit). Nous avons utilisé 70 nœuds dans nos simulations. Ce choix est fait pour faciliter la visualisation.

Rappelons que le modèle de diffusion de données est piloté par les événements. Les nœuds détectent des valeurs telles que la température, la pression, le son, etc. Lorsque ces valeurs atteignent un certain seuil, elles sont transmises au Sink(puit).

Concernant le routage, nous avons effectué notre analyse en utilisant AODV. Notre choix est justifié par son adéquation avec RCSF. En plus de la capacité à soutenir la mobilité.

Plus de détails sur les configurations ns-2 sont présentés à l'annexe A de la section 1 à la section 4.

3.3.2 Les attaques simulées :

Nous avons simulé le comportement normal et trois comportements malveillants différents: Blackhole, Hello Flood et DoS.

Concernant le comportement Normal, nous avons choisi le protocole AODV pour le routage et le protocole TCP pour la transmission. Il y a 70 nœuds dans notre topologie, alors que le plan de livraison est à un moment donné, un nœud envoie des données au sink (nœud 0). Le nœud commence à envoyer des données au hasard, ce qui nous permet de simuler des applications réelles. Le script TCL ns-2 du comportement normal est présenté à l'annexe A, section 8.

Pour reproduire le même scénario Normal lors de la simulation d'attaques, nous avons sélectionné le scénario (Sauvegarde de l'identité des nœuds de travail et de leur temps de début de transmission de données). Le script TCL ns-2 à exécuter est détaillé à l'annexe A, section 5.

En ce qui concerne les comportements malveillants, nous avons généré la même topologie que le comportement normal avec les mêmes paramètres, puis nous chargeons le fichier qui contient les identités des nœuds et leur temps de départ des données d'envoi. Ensuite, nous avons choisi au hasard un seul nœud pour qu'il agisse de manière malveillante. Nous avons choisi trois comportements malveillants différents:

- Blackhole : Le nœud malveillant supprime tous les paquets qui le traversent. Pour ce faire, il attire ses nœuds voisins en forgeant une réponse d'itinéraire avec moins de nombres de sauts et un plus grand nombre de séquences. Nous avons simulé cela en ajoutant cet acte malveillant au protocole AODV, comme indiqué à l'annexe A, section 9.1, et au script TCL, section 9.2.

- Hello Flood : Le nœud flooder continue d'envoyer des messages RREQ malgré la réception de messages RREP, dans le but de gaspiller la bande passante du réseau et d'épuiser ses ressources. Nous avons simulé ceci en ajoutant le code indiqué à l'annexe A, section 10.1, au protocole AODV et le script TCL est détaillé à la section 10.2.

- DoS : Le nœud attaquant continue d'envoyer des paquets au récepteur, dans le but de le rendre inactif. Le script TCL est présenté à l'annexe A, section 11

3.3.3 Les fichiers trace(.tr) :

Le fichier écrit par une application pour stocker les informations de réseau global informations et Dans NS2, il est appelé fichier de trace.

```
s -t 2.148582125 -Hs 66 -Hd -2 -Ni 66 -Nx 70.00 -Ny 160.00 -Nz 0.00 -Ne 10.000000 -Nl
AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 66.0 -Id 0.2 -It tcp -Il 40 -If 3 -Ii 0 -Iv 32 -
Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
r -t 2.148582125 -Hs 66 -Hd -2 -Ni 66 -Nx 70.00 -Ny 160.00 -Nz 0.00 -Ne 10.000000 -Nl
RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 66.0 -Id 0.2 -It tcp -Il 40 -If 3 -Ii 0 -Iv 32 -
Pn tcp -Ps 0 -Pa 0 -Pf 0 -Po 0
s -t 2.148582125 -Hs 66 -Hd -2 -Ni 66 -Nx 70.00 -Ny 160.00 -Nz 0.00 -Ne 10.000000 -Nl
RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 66.255 -Id -1.255 -It AODV -Il 48 -If 0 -Ii 0 -
Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 0 -Pds 0 -Ps 66 -Pss 4 -Pc REQUEST
N -t 2.148857 -n 57 -e 9.892505
N -t 2.148857 -n 65 -e 9.892505
N -t 2.148857 -n 67 -e 9.892505
r -t 2.149402191 -Hs 57 -Hd -2 -Ni 57 -Nx 70.00 -Ny 140.00 -Nz 0.00 -Ne 9.892505 -Nl
RTR -Nw --- -Ma 0 -Md ffffffff -Ms 42 -Mt 800 -Is 66.255 -Id -1.255 -It AODV -Il 48 -If
0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 0 -Pds 0 -Ps 66 -Pss 4 -Pc REQUEST
r -t 2.149402191 -Hs 65 -Hd -2 -Ni 65 -Nx 50.00 -Ny 160.00 -Nz 0.00 -Ne 9.892505 -Nl
RTR -Nw --- -Ma 0 -Md ffffffff -Ms 42 -Mt 800 -Is 66.255 -Id -1.255 -It AODV -Il 48 -If
0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 0 -Pds 0 -Ps 66 -Pss 4 -Pc REQUEST
r -t 2.149402191 -Hs 67 -Hd -2 -Ni 67 -Nx 90.00 -Ny 160.00 -Nz 0.00 -Ne 9.892505 -Nl
RTR -Nw --- -Ma 0 -Md ffffffff -Ms 42 -Mt 800 -Is 66.255 -Id -1.255 -It AODV -Il 48 -If
0 -Ii 0 -Iv 30 -P aodv -Pt 0x2 -Ph 1 -Pb 1 -Pd 0 -Pds 0 -Ps 66 -Pss 4 -Pc REQUEST
s -t 2.152492982 -Hs 65 -Hd -2 -Ni 65 -Nx 50.00 -Ny 160.00 -Nz 0.00 -Ne 9.892505 -Nl
RTR -Nw --- -Ma 0 -Md ffffffff -Ms 42 -Mt 800 -Is 65.255 -Id -1.255 -It AODV -Il 48 -If
0 -Ii 0 -Iv 29 -P aodv -Pt 0x2 -Ph 2 -Pb 1 -Pd 0 -Pds 0 -Ps 66 -Pss 4 -Pc REQUEST
N -t 2.152608 -n 56 -e 9.892318
N -t 2.152608 -n 64 -e 9.892318
N -t 2.152608 -n 66 -e 9.892172
r -t 2.153153049 -Hs 56 -Hd -2 -Ni 56 -Nx 50.00 -Ny 140.00 -Nz 0.00 -Ne 9.892318 -Nl
RTR -Nw --- -Ma 0 -Md ffffffff -Ms 41 -Mt 800 -Is 65.255 -Id -1.255 -It AODV -Il 48 -If
0 -Ii 0 -Iv 29 -P aodv -Pt 0x2 -Ph 2 -Pb 1 -Pd 0 -Pds 0 -Ps 66 -Pss 4 -Pc REQUEST
r -t 2.153153049 -Hs 64 -Hd -2 -Ni 64 -Nx 30.00 -Ny 160.00 -Nz 0.00 -Ne 9.892318 -Nl
RTR -Nw --- -Ma 0 -Md ffffffff -Ms 41 -Mt 800 -Is 65.255 -Id -1.255 -It AODV -Il 48 -If
0 -Ii 0 -Iv 29 -P aodv -Pt 0x2 -Ph 2 -Pb 1 -Pd 0 -Pds 0 -Ps 66 -Pss 4 -Pc REQUEST
r -t 2.153153049 -Hs 66 -Hd -2 -Ni 66 -Nx 70.00 -Ny 160.00 -Nz 0.00 -Ne 9.892172 -Nl
RTR -Nw --- -Ma 0 -Md ffffffff -Ms 41 -Mt 800 -Is 65.255 -Id -1.255 -It AODV -Il 48 -If
```

Figure III. 3: Capture de fichier trace.

La liste suivante montre les variables du fichier de trace et ses explications :

s - send packet

r - received packet

d - packet dropped

f - packet forwarded

c - collision of packet at MAC level

t - time at which packet tracing started

Hs - ID of the hop

Hd - ID of the next hop towards destination

Ni - Node ID

Nx,Ny,Nz - Co ordinates that the nodes situated

Ne - Node energy level

Nl - Trace level

Nw - Reason of the event

AGT - Agent

RTR -Routing

END - DROP End of Simulation

COL - DROP MAC COLLISION

DUP - DROP MAC DUPLICATE

DERR - DROP MAC PACKET ERROR

RET - DROP MAC RETRY COUNT EXCEED

STA - DROP MAC INVALID STATE

BSY - DROP MAC BUSY

NRTE - DROP RTR - NO ROUTE

LOOP - DROP RTR ROUTE LOOP

TTL - DROP RTR TTL has reached Zero

TOUT - DROP-RTR-QTIME OUT Expired

Is - Source address of source port

Id - Destination address of destination port

Il - Packet Size

If - Flow ID

Ii - Unique ID

Iv - TTL value next hop

Nous avons lancé les simulations. Cela a conduit à un ensemble de fichiers de trace à partir desquels nous avons extrait les attributs ciblés. Nous avons déployé des scripts AWK pour effectuer cette extraction. Le script est reporté en annexe B.

3.3.4 Les paramètres de simulations :

Pour prouver l'efficacité de notre méthode pour détecter une attaque DOS et évaluer sa performance, nous avons utilisé les paramètres de simulation listés dans le tableau 3.2 .

Paramètre	Signification
Le nombre des nœuds	70
Taille de réseau	200*200 m²
Porté de communication	25m
Taille de paquet	1024 bits
Protocole de routage	AODV
Temps de simulation	80 s

Tableau III. 2: Paramètres de simulation**.**

3.3.5 Les attributs collectés :

Il existe de nombreux attributs pour caractériser un comportement au niveau de la connexion. Dans notre étude, nous avons réparties en 3 catégories: temps, routage et données comme indiqué dans le tableau 3.3.

Attributs	Types	La description
Durée	Temps	La durée entre l'envoi du premier paquet et la réception du dernier paquet.
Toutes les données de paquets reçus	Données, routage	Tous les paquets (indépendamment de qui les a envoyés et de leur type) que le récepteur a reçus pendant la durée de cette connexion
Demande de route AODV	Routage	Nombre de paquets AODV RREQ envoyés par le nœud expéditeur.
AODV Route Réponse	Routage	Le nombre de paquets RREP AODV envoyés par le nœud émetteur.
Paquet TCP envoyé	Données	Le nombre de paquets TCP envoyés par le noeud expéditeur.
Paquet TCP reçu	Données	Le nombre de paquets TCP reçus par le récepteur.
Le paquet TCP a chuté	Données	Le nombre de paquets TCP qui ont été supprimés.
Paquet TCP forward	Données	La somme du nombre de fois que chaque paquet TCP dans cette connexion a été transféré.
Énergie consommée		La quantité d'énergie consommée par tous les nœuds inhérents à la connexion.
Taux de livraison de paquets	Données	le rapport des paquets TCP reçus aux paquets TCP envoyés.
Délai moyen	Temps	Le délai moyen des paquets TCP reçus pendant la connexion.
Max hop	Routage	Nombre maximal de sauts pendant la connexion.
Houblon moyen	Routage	Nombre moyen de sauts pendant la connexion.
Débit	Données	la quantité de données

	reçues dans la connexion durée (kbps).

Tableau III. 3: Descriptions des attributs collectés.

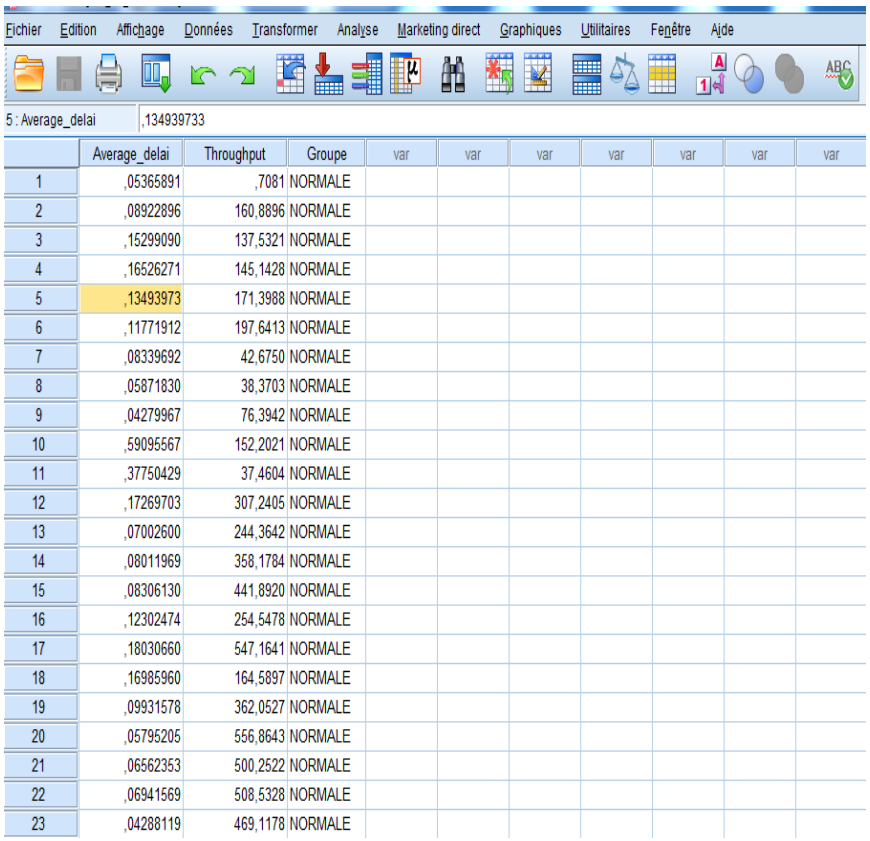
-D'après le tableau précédent, nous notons que les attributs (débit et délai moyen) sont les plus importantes dans notre étude et ces derniers sont nommés les attributs ciblés.

4 MANOVA avec SPSS

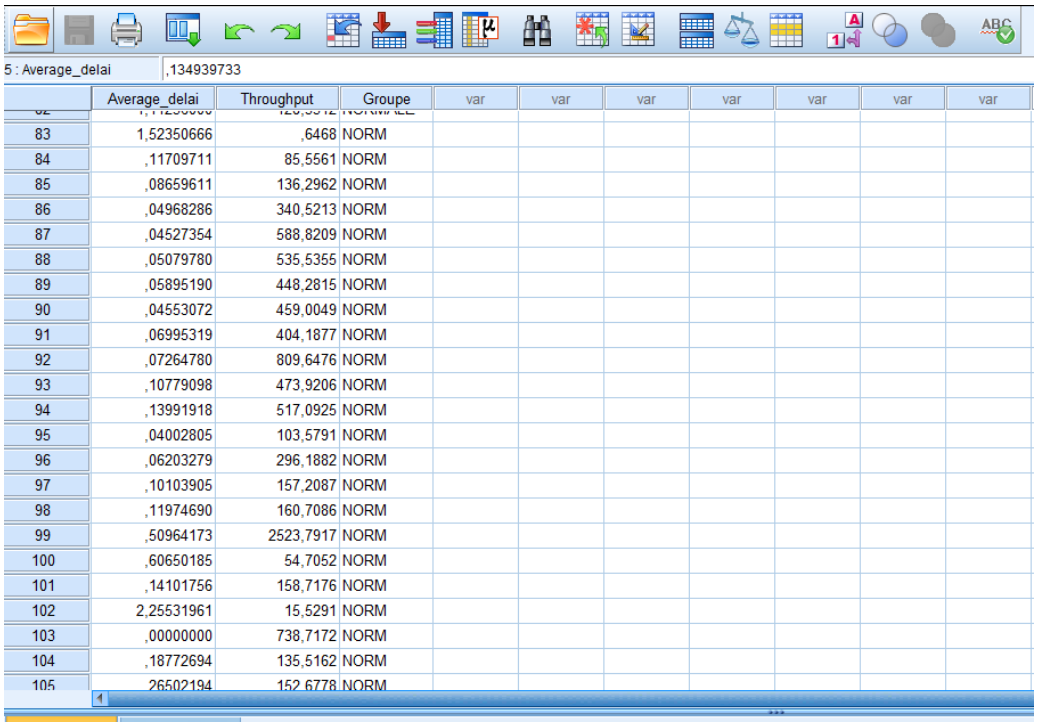
Nous avons importé des données normales et des données anomalies liées aux attaques DOS via NS2, et nous appliquerons MANOVA à ces données via le programme SPSS.

4.1 Cas normale :

Importation des données comme montre la figure :



	Average_delai	Throughput	Groupe	var	var	var	var	var	var	var
1	,05365891	,7081	NORMALE							
2	,08922896	160,8896	NORMALE							
3	,15299090	137,5321	NORMALE							
4	,16526271	145,1428	NORMALE							
5	,13493973	171,3988	NORMALE							
6	,11771912	197,6413	NORMALE							
7	,08339692	42,6750	NORMALE							
8	,05871830	38,3703	NORMALE							
9	,04279967	76,3942	NORMALE							
10	,59095567	152,2021	NORMALE							
11	,37750429	37,4604	NORMALE							
12	,17269703	307,2405	NORMALE							
13	,07002600	244,3642	NORMALE							
14	,08011969	358,1784	NORMALE							
15	,08306130	441,8920	NORMALE							
16	,12302474	254,5478	NORMALE							
17	,18030660	547,1641	NORMALE							
18	,16985960	164,5897	NORMALE							
19	,09931578	362,0527	NORMALE							
20	,05795205	556,8643	NORMALE							
21	,06562353	500,2522	NORMALE							
22	,06941569	508,5328	NORMALE							
23	,04288119	469,1178	NORMALE							



5 : Average_delai ,134939733

	Average_delai	Throughput	Groupe	var	var	var	var	var	var	var
83	1,52350666	,6468	NORM							
84	,11709711	85,5561	NORM							
85	,08659611	136,2962	NORM							
86	,04968286	340,5213	NORM							
87	,04527354	588,8209	NORM							
88	,05079780	535,5355	NORM							
89	,05895190	448,2815	NORM							
90	,04553072	459,0049	NORM							
91	,06995319	404,1877	NORM							
92	,07264780	809,6476	NORM							
93	,10779098	473,9206	NORM							
94	,13991918	517,0925	NORM							
95	,04002805	103,5791	NORM							
96	,06203279	296,1882	NORM							
97	,10103905	157,2087	NORM							
98	,11974690	160,7086	NORM							
99	,50964173	2523,7917	NORM							
100	,60650185	54,7052	NORM							
101	,14101756	158,7176	NORM							
102	2,25531961	15,5291	NORM							
103	,00000000	738,7172	NORM							
104	,18772694	135,5162	NORM							
105	,26502194	152,6778	NORM							

Figure III. 4 :capture des données dans le cas normale

-application analyse multivarié MANOVA sur les deux groupes des donnés normaux

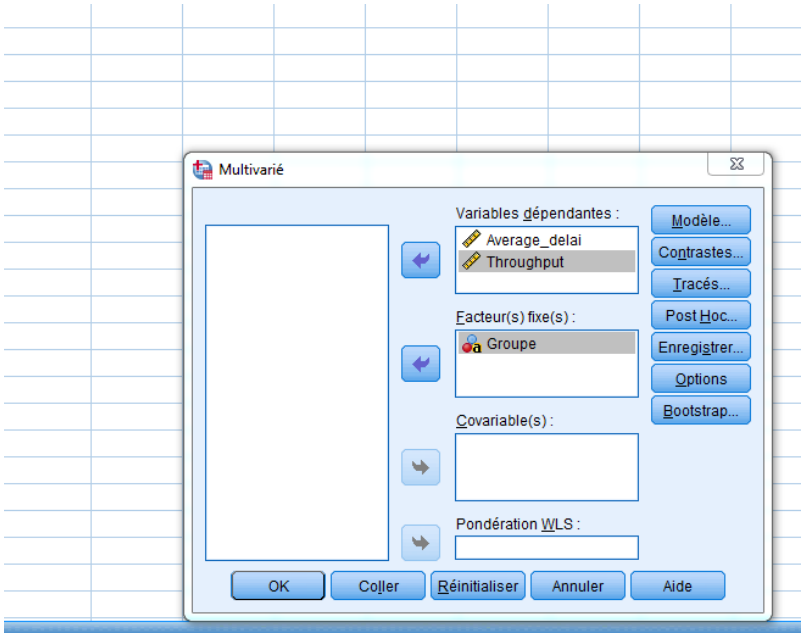


Figure III. 5 :capture de paramètres MANOVA

- Les résultats :

Statistiques descriptives

	Groupe	Moyenne	Ecart type
Average_delai	NORM	,2288718312	,4731345314
	NORMALE	,1992412148	,2488104100
	Total	,2139656315	,3764577845
Throughput	NORM	331,496198	317,9374930
	NORMALE	323,618693	372,2560411
	Total	327,533281	345,2839946

Figure III. 6 : capture des statistiques descriptives dans le cas normale.

On remarque que les moyennes des deux groupes sont similaires .

Nous passons à le test multivarié :

Tests multivariés[a]

Effet		Valeur	F	ddl de l'hypothèse	Erreur ddl	Signification	Eta-carré partiel	Paramètre de non-centralité	Puissance observée[c]
Constante	Trace de Pillai	,592	116,306[b]	2,000	160,000	,000	,592	232,612	1,000
	Lambda de Wilks	,408	116,306[b]	2,000	160,000	,000	,592	232,612	1,000
	Trace de Hotelling	1,454	116,306[b]	2,000	160,000	,000	,592	232,612	1,000
	Plus grande racine de Roy	1,454	116,306[b]	2,000	160,000	,000	,592	232,612	1,000
Groupe	Trace de Pillai	,002	,152[b]	2,000	160,000	,859	,002	,303	,073
	Lambda de Wilks	,998	,152[b]	2,000	160,000	,859	,002	,303	,073
	Trace de Hotelling	,002	,152[b]	2,000	160,000	,859	,002	,303	,073
	Plus grande racine de Roy	,002	,152[b]	2,000	160,000	,859	,002	,303	,073

a. Plan : Constante + Groupe

b. Statistique exacte

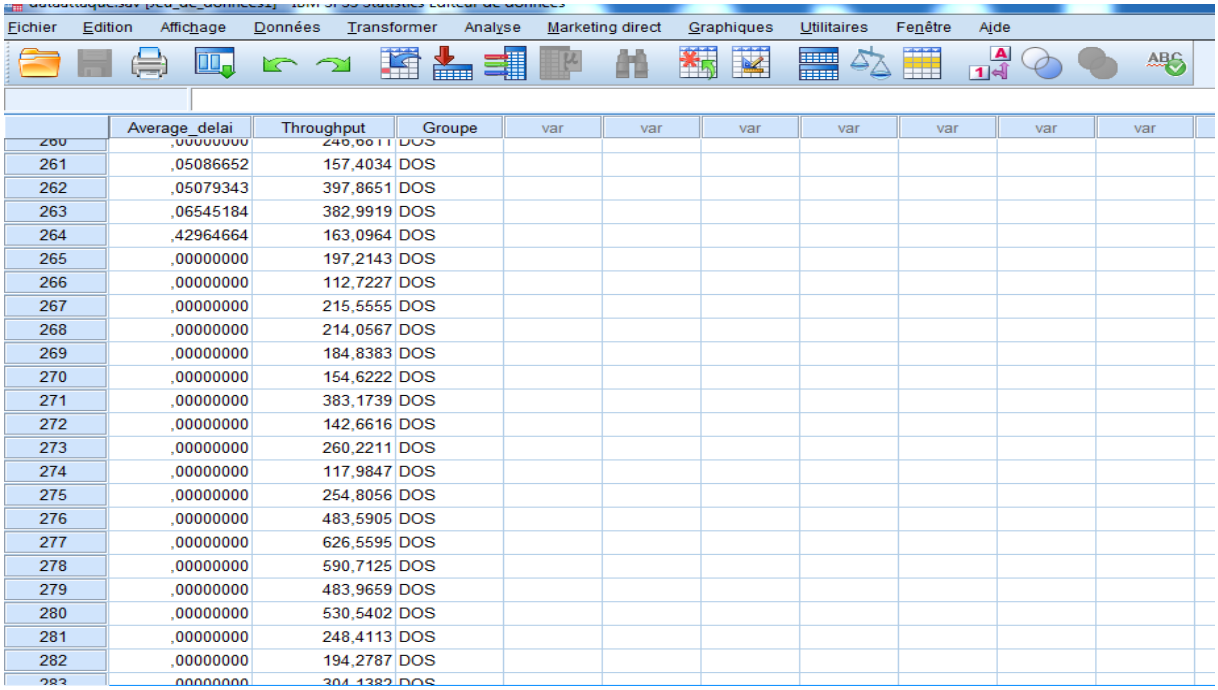
c. Calcul à l'aide d'alpha = ,05

Figure III. 7 :capture des testes multivariés dans le cas normale.

Dans les quatre tests la valeur p de signification supérieure à alpha=0.05 donc il y’a aucune déférence entre les deux groupes des données ,on conclure que Il n'y a pas d'attaque.

4.2 Cas attaque DOS :

Nous apportons les données normales et les comparons avec les données inhabituelles générées par les attaques DOS.



	Average_delai	Throughput	Groupe	var	var	var	var	var	var	var
260	,00000000	246,6811	DOS							
261	,05086652	157,4034	DOS							
262	,05079343	397,8651	DOS							
263	,06545184	382,9919	DOS							
264	,42964664	163,0964	DOS							
265	,00000000	197,2143	DOS							
266	,00000000	112,7227	DOS							
267	,00000000	215,5555	DOS							
268	,00000000	214,0567	DOS							
269	,00000000	184,8383	DOS							
270	,00000000	154,6222	DOS							
271	,00000000	383,1739	DOS							
272	,00000000	142,6616	DOS							
273	,00000000	260,2211	DOS							
274	,00000000	117,9847	DOS							
275	,00000000	254,8056	DOS							
276	,00000000	483,5905	DOS							
277	,00000000	626,5595	DOS							
278	,00000000	590,7125	DOS							
279	,00000000	483,9659	DOS							
280	,00000000	530,5402	DOS							
281	,00000000	248,4113	DOS							
282	,00000000	194,2787	DOS							
283	,00000000	304,1382	DOS							

Figure III. 8:capture des données dans le cas DOS.

-application analyse multivarié MANOVA sur les deux groupes normale et les données anomalies.

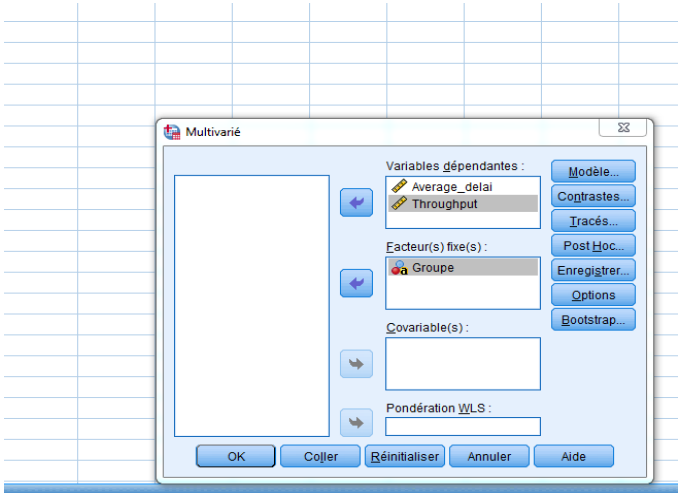


Figure III. 9 :capture des paramètres de MANOVA.

-résultats :

Statistiques descriptives

	Groupe	Moyenne	Ecart type	N
Average_delai	DOS	,0068900205	,0479006545	303
	NORMALE	,1707697045	,4136307000	214
	Total	,0747241644	,2801699301	517
Throughput	DOS	220,469804	136,6616236	303
	NORMALE	175,608733	234,5644244	214
	Total	201,900618	184,7481090	517

Figure III. 10:capture des statistiques descriptives dans le cas DOS.

-on remarque que le délai moyenne dans le cas DOS est moins que Normale Parce que dans ce cas, le taux de transmission est élevé donc le débit augmente.

Nous passons à le test multivarié :

Tests multivariés[a]

Effet		Valeur	F	ddl de l'hypothèse	Erreur ddl	Signification	Eta-carré partiel	Paramètre de non-centralité	Puissance observée[c]
Constante	Trace de Pillai	,569	338,929[b]	2,000	514,000	,000	,569	677,857	1,000
	Lambda de Wilks	,431	338,929[b]	2,000	514,000	,000	,569	677,857	1,000
	Trace de Hotelling	1,319	338,929[b]	2,000	514,000	,000	,569	677,857	1,000
	Plus grande racine de Roy	1,319	338,929[b]	2,000	514,000	,000	,569	677,857	1,000
Groupe	Trace de Pillai	,090	25,493[b]	2,000	514,000	,000	,090	50,985	1,000
	Lambda de Wilks	,910	25,493[b]	2,000	514,000	,000	,090	50,985	1,000
	Trace de Hotelling	,099	25,493[b]	2,000	514,000	,000	,090	50,985	1,000
	Plus grande racine de Roy	,099	25,493[b]	2,000	514,000	,000	,090	50,985	1,000

a. Plan : Constante + Groupe

b. Statistique exacte

c. Calcul à l'aide d'alpha = ,05

Figure III. 11 :capture des testes multivariés dans le cas attaque DOS.

Dans les quatre tests la valeur p de signification inférieure à alpha=0.05 donc il y'a déférence entre les deux groupes des données ,on conclure que il existe un d'attaque.

Donc IDS découverte un trafic suspect ,il fait un alarme.

5 Script R détection

Dans cette partie nous avons présenté notre script de détection programmé avec le langage R, ce script est illustré dans la figure III.12.

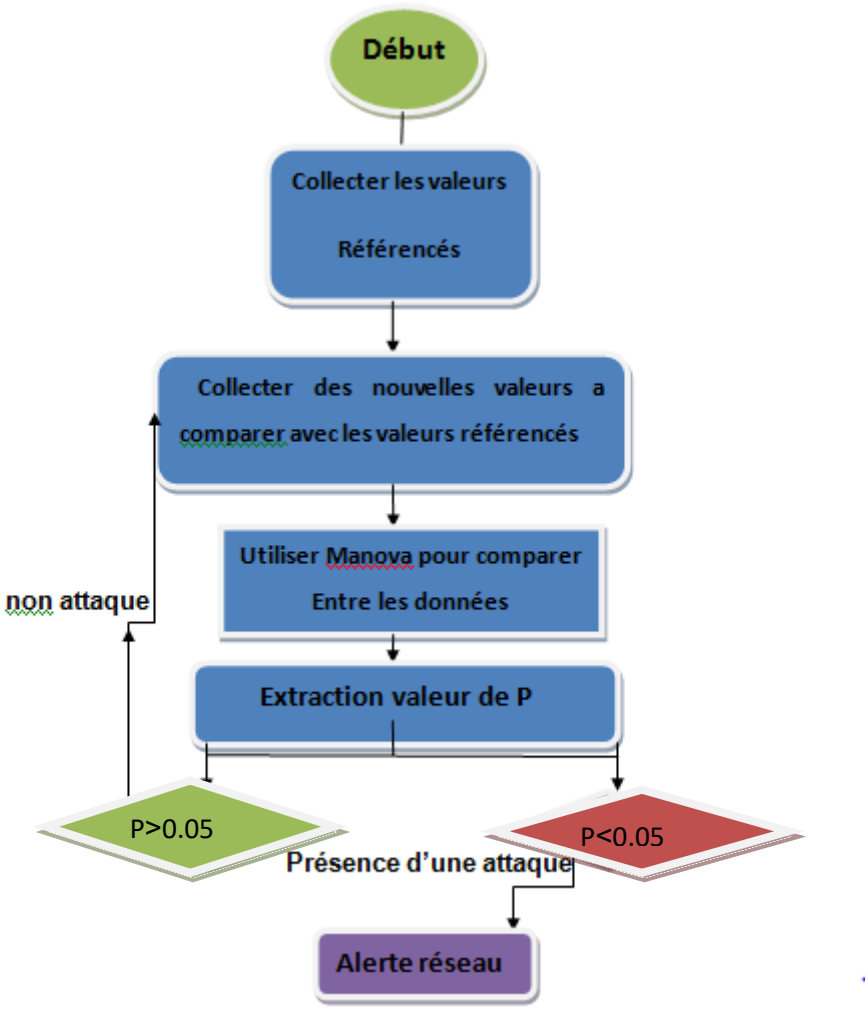


Figure III. 12 : Fonctionnement Script R de détection

6 Résultats et interprétations

6.1 Métriques :

L'évaluation de la qualité de notre système IDS est faite avec les quatre éléments de base qui sont : les vrais positifs (VP), i.e. les anomalies effectivement détectées, les faux négatifs (FN), i.e. les anomalies non-détectées, les faux positifs (FP), i.e. les fausses alarmes et les vrais négatifs (VN). On peut les résumer sous la forme d'une matrice de confusion (Table 3.3).

	Prédit positif	Prédit négatif
Vrais positifs	VP	FN
Vrais négatifs	FP	VN

Tableau III. 4;: Matrice de confusion pour un problème de classification à 2 classes.

Pour mesurer le pourcentage d'échantillons correctement classés, on utilise l'incertitude (1) (i.e. le ratio de vrais positifs et vrais négatifs sur le nombre total d'échantillons). Cependant, il peut fausser l'évaluation lorsque le nombre de vrais négatif est considérablement supérieur au nombre de vrais positifs. C'est pourquoi l'on utilise d'autres métriques en complément.

$$\textbf{Incertitude : I} = \frac{\textbf{VP+VN}}{\textbf{VP+VN+FP+FN}} \quad \textbf{(1)}$$

Le deuxième critère est la précision (2), i.e. le ratio entre le nombre de varies anomalies et le nombre d'échantillons classés par l'algorithme comme des anomalies. Plus la valeur est petite et plus il y a de faux positifs. Le second critère est le rappel (3), qui est le pourcentage d'anomalies correctement classées en tant que telles parmi toutes les vraies anomalies. Ainsi, plus le rappel est petit et plus il y a d'anomalies non détectées.

$$\textbf{Précision : } P = \frac{VP}{VP + FP} (2)$$

$$\textbf{Rappel: } R = \frac{VP}{VP + FN} \quad (3)$$

6.2 Résultats et analyse :

Sur la base de 30 simulations déployées, nous avons obtenu des résultats qu'ils sont présentés dans les tables suivantes :

Blackhole :

	Prédit positif	Prédit négatif
Vrais positifs	35%	16%
Vrais négatifs	24%	25%

Tableau III. 5:Taux de détection de l'attaque Blackhole.

Dos :

	Prédit positif	Prédit négatif
Vrais positifs	44%	17%
Vrais négatifs	13%	26%

Tableau III. 6: Taux de détection de l'attaque DoS.

Flood :

	Prédit positif	Prédit négatif
Vrais positifs	45%	18%
Vrais négatifs	17%	20%

Tableau III. 7 : Taux de détection de l'attaque Flood.

Métriques / Attaques	VP	VN	FP	FN	Incertitude	Précision	Rappel
Blackhole	35%	25%	16%	24%	60%	68%	59%
Dos	44%	26%	13%	17%	70%	77%	72%
Flood	45%	20%	17%	18%	65%	72%	71%

Tableau III. 8: Performance et précision.

Nous avons atteint 68% de précision dans le scénario de Blackhole, 72% de précision dans le scénario Hello-Flood, et 77% de précision dans le scénario DoS. Les performances de notre système sont plutôt acceptables, mais pas assez fiables parce que nous pensons qu'il existe des attributs que nous devons prendre en considération afin d'avoir une précision suffisante.

7 Conclusion

Dans ce chapitre, nous avons présenté notre système IDS basé sur un modèle statistique (MANOVA) utilisé pour comparer les variances d'échantillons pour RCSF, et nous avons présenté aussi l'ensemble des données est le résultat des données de simulation de prétraitement. Nous avons justifié tous les choix effectués sur le niveau de caractérisation des attaques, les attributs ciblés et l'algorithme à utiliser. Nous avons décrit notre étude et commenté les résultats donnés.

Conclusion générale :

Les attaques par déni de services existent depuis de nombreuses années et sont parfois médiatisées. Elles ont su se développer au fur et à mesure du développement des réseaux, tout en étant simples à mettre en œuvre. Les RCSF souffrent aussi de ce type d'attaque, et bien qu'aujourd'hui, il existe des méthodes de détections et de protection contre le déni de service, ils sont toujours complexes à mettre en place avec une efficacité pas toujours optimale et ceci à cause des spécificités des RCSF.

Dans ce travail, nous avons traité la problématique de la détection d'intrusion dans les (RCSF), nous avons également mentionné comment il est important de sécuriser ces réseaux face aux attaques DoS. Notre objectif est de déployer une solution allégée qui offre de hautes performances tout en respectant les spécificités et les limites des RCSF en termes d'énergie, de mémoire et de puissance de calcul.

La solution consistée à détecter les attaques par l'étude de l'impact des attaques DoS sur la variance d'un ou plusieurs attributs d'un RCSF. Pour ce faire, nous avons expérimenté de nombreux scénarios de simulation contenant plusieurs comportements ciblés. Ces comportements sont Normal, Blackhole, Hello-Flood et DoS. Ensuite on a déterminé quels sont les attributs les plus sensibles aux attaques DoS, pour les utiliser comme des indices révélateurs d'attaques DoS. Plus tard nous avons appliqué notre méthode de détection d'intrusion (IDS) qui est basée sur l'analyse multivarié de la variance MANOVA. Les performances obtenues sont plutôt acceptables, mais nécessitent des améliorations et des ajustements que nous recommandons pour des travaux ultérieurs.

Finalement, ce travail peut être enrichi en essayant d'autres configurations et d'autres protocoles de simulation.

Annexes

Annexe A :

Comportements simulés NS2 :

1. Configuration et paramètres des nœuds

Set val (chan)	channel/wireless channel	; # type de canal
Set val (prop)	Propagation /TwoRayGround	; # modèle de propagation radio
Set val (netif)	Phy/ Wireless Phy	#type d'interface réseau
Set val (mac)	Mac/802_1 1	; # Type MAC
Set val (ifq)	Queue/ DropTail / PriQueue	;# type de file d'attente
Set val (ll)	LL	;# type de couche de liaison
Set val (ant)	Antenna /OmniAntenna	;# modèle d'antenne
Set val (ifqlen)	200	;# paquet maximum en ifq
Set val (nn)	70	;# nombre de nœuds mobiles
Set val (rp)	AODV	;# type de protocole
Set val (x)	200	;# X dimension de la topographie
Set val (y)	200	;# Y dimension de la topographie
Set val (stop)	80	;# période de simulation

2. Initialisation d'objets et de fichiers de trace

```
Set ns                    [new Simulator]
Set tracefd   [open Normal.tr w]
Set namtrace   [open normal.nam w]
# configurer un nouveau format de fichier de trace
$ns use-newtrace
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val (x) $val( y )
```

Set dist (25m) 3.07645 e-07

Phy/WirelessPhy set CSThresh_ $ dist (25m)

Phy/WirelessPhy set RXThresh_ $ dist (25m)

3. Définition de la topographie et des valeurs aux paramètres configurés

 # configurer un objet de topographie

Set topo [new Topography]

$topo load_flatgrid $val (x) $val (y)

#Création du directeur général des opérations

create-god $val (nn)

 # configurer les noeuds

$ns node-config-adhocRouting $val (rp) \

-llType $val (ll) \

-macType $val (mac) \

-ifqType $val (ifq) \

-ifqLen $val (ifqlen) \

-antType $val (ant) \

-propType $val (prop) \

-phyType $val (netif) \

-channel [new $val (chan)] \

-topoInstanc e $topo \

-energyModel "EnergyModel" \

-initialEnergy 10 \

-txPower 0.33 \

-rxPower 0.1 \

-idlePower 0.05 \

-sleepPower 0.03 \

-agentTrace ON \

-routerTrace ON \

-macTrace OFF \

-movementTrace OFF \

4. Création et positionnement de nœuds

for {set i 0} {$i< $val (nn)} {incri} {

Set mnode_ ($i) [$ns node]

$mnode_ ($i) set X_ [expr [expr [expr $i * 20] % 180] + 10]

```
$mnode_ ($i) set Y_ [expr [expr [expr $i * 70] / 280] * 10]
$mnode_ ($i) set Z_ 0.0}
$mnode (0) label " Sink "
for {set i 0} {$i< $val (nn)} {incri} {
$ns initial_node_pos $mnode ($i) 10}
```

5. Enregistrement du scénario de comportement normal

```
for {set i 0} {$i< $wn} {incri} {
Set random_node [expr int (rand ( ) _ 69) + 1]
#ajouter les nœuds dans les fichiers
puts $ nodes file "$random_node"

#Configurer une connexion TCP
Set tcp ($i) [new Agent/TCP]
$tcp ($i) set class 2
$tcp ($i) set {id [expr $ i + 1]
Set sink ($i) [new Agent/TCPSink]
$ns attach-agent $mnode ($random_node) $tcp_ ($i)
$ns attach-agent $mnode (0) $sink ($i)
$ns connect $tcp ($i) $sink ($i)
Set ftp ($i) [new Application /FTP]
$ftp ($i) attach-agent $tcp ($i)}

for {set i 0} {$i< $wn} {incri} {
Setstart_time [expr (rand ( ) + 2)]
Set stop_time [expr $start_time + 77]
puts $timefile" $start_time "
$ns at $start_time " $ftp ($i) start"
$ns at $stop_time " $ftp ($i) stop "}
```

6. Chargement du scénario de comportement normal

```
foreach line $nf {
#Configureruneconnexion TCP
set tcp ($j) [new Agent/TCP]
$tcp ($j) set class 2
$tcp ($j) set fid [expr $ j + 1]
set sink ($j) [new Agent/TCPSink]
$ns attach-agent $mnode($line) $tcp ($j)
$ns attach-agent $mnode(0) $sink ($j)
$ns connect $tcp ($j) $sink ($j)
```

```
if {$random_node == $line} {
set random_node [expr int (rand ( ) _ 69 ) + 1]}
set ftp_($j) [new Application/FTP]
$ftp_($j) attach-agent $tcp_($j)
incrj}
set j 0
foreach line $tf {
set start_time $line
set stop_time [expr $start_time + 77]
$ns at $start_time " $ftp ($j) start "
$ns at $stop_time " $ftp ($j) stop "
incr j}
```

7. Terminer la simulation et fermer les fichiers utilisés

```
# Avertir les noeuds lorsque la simulation est terminée
for {set i 0} {$i< $val (nn)} {incri} {
$ns at $val (stop) "$mnode_($i) reset ; "}
# fin nam et la simulation
$ns at $val (stop) " $ns nam-end-wireless $val (stop)"
$ns at [expr $val (stop) + 0.01] " puts \"end simulationn\" ; $ns halt "
proc stop {} {
global ns tracefdnamtrace
$ns flush-trace
close $ timefile
close $ nodesfile
close $ tracefd
close $namtrace}
```

8. Comportement normal

```
//Configuration et paramètres des nœuds
#définir le nombre des nœuds qui enverraient des données
Set wn [lindex $argv 0]
# Ouverture de fichiers en mode écriture pour sauvegarder les temps de démarrage et l'envoi
Set time file [open timef.txt w]
Set nodes file [open nodes.txt w]
// Initialisation d'objets et les fichiers trace
// Définition de la topographie et des valeurs aux paramètres configurés
// Création et positionnement de nœuds
// Enregistrement du scénario de comportement normal
//Terminer la simulation et fermer les fichiers utilisés
```

9. Blackholeattack

a. Le code ajouté à AODV.h

// Dans AODV classe agent de routage

boolmalicious ;

// Reste du code est ajouté à AODV.cc

9.2 TCL script

// Configuration et paramètres des nœuds

 # définir le nombre des nœuds qui enverraient des données

Set wn [lindex $argv 0]

 # Ouverture de fichiers en mode écriture pour sauvegarder les temps de démarrage et l'envoi

set time file [open timef.txt r]

set nodes file [open nodes.txt r]

set tf [read $ time file]

set nf [read $ nodes file]

set j 0

// Initialisation d'objets et les fichiers trace

// Définition de la topographie et des valeurs aux paramètres configurés

// Création et positionnement de nœuds

// Chargement scénario de comportement normal

set random_node [expr int (rand () _69) + 1]

$ns at 40.0 " [$mnode ($random_node) set ragent_] blackhole"

puts "$random_node"

// Terminer la simulation et fermer les fichiers utilisés

10. Hello flood attack

10.1. Le code ajouté à AODV.h

#définir FLOOD_INTERVAL 0.09

// dans les minuteries

class FloodTimer : public Handler{

public :

FloodTimer (AODV* a) : agent (a) {}

void handle (Event*) ;

private :

AODV *agent ;

Event intr;} ;

// Dans AODV classe agent de routage

friend class FloodTimer ;

// dans les routines TX paquet

voidFloodRREQ(nsaddr_t dst) ;

// Dans AODV: Agent Timers

boolflooder ;

FloodTimerftimer ;

// Reste du code est ajouté à AODV.cc

10.2 TCL script

// Configuration et paramètres des nœuds

 # définir le nombre des nœuds qui enverraient des données

setwn [lindex $argv 0]

 # Ouverture de fichiers en mode écriture pour sauvegarder les temps de démarrage et l'envoi

set time file [open timef.txt r]

set nodes file [open nodes.txt r]

set tf [read $ time file]

set nf [read $ nodes file]

set j 0

// Initialisation d'objets et les fichiers trace

// Définition de la topographie et des valeurs aux paramètres configurés

// Création et positionnement de nœuds

// Chargement scénario de comportement normal

set random_node [expr int (rand () * 69) + 1]

$ns at 40.0" [$mnode ($random_node) set ragent] flooder"

puts "$random_node"

// Terminer la simulation et fermer les fichiers utilisés

11. DoSattack

// Configuration et paramètres des nœuds

 # définir le nombre des nœuds qui enverraient des données

setwn [lindex $argv 0]

 # Ouverture de fichiers en mode écriture pour sauvegarder les temps de démarrage et l'envoi

set time file [open timef.txt r]

set nodes file [open nodes.txt r]

set tf [read $ time file]

set nf [read $ nodes file]

set j 0

// Initialisation d'objets et les fichiers trace

// Définition de la topographie et des valeurs aux paramètres configurés

// Création et positionnement de nœuds

// Chargement scénario de comportement normal

 # créer une attaque DoS

```
set udp [new Agent/UDP]
$udp set fid $j
set null [new Agent/Nul l]
$ns attach-agent $mnode($random_node) $udp
$ns attach-agent $mnode(0) $null
set cbr [ new Application / Traffic /CBR]
$cbr set inter val 0.001
$cbr set random_ 1
$cbr set packetSize 1000
$cbr set maxpkts 300000
$cbr attach-agent $udp
$ns connect $udp $null
$ns at 40.0 " $cbr start "
$ns at 80.0 " $cbr stop "
// Terminer la simulation et fermer les fichiers utilisés
```

Annexe B

Script AWK

```
BEGIN {

i=0;
j=0;
n=0;
inc=0;
for(i=1;i< 70;i++){
start_time[i] = 100.000000000;}
}

{
if (FILENAME == "nodes.txt")
{
cnxs++;
sender_node[cnxs] = $0;
}
else
{
state      =    $1;
time       =    $3;
flow       =    $39;
pkt_size   =    $37;
node_id    =    $9;
level           =    $19;
pkt_type   =    $35;
packet_id  =    $41;

if(state != "N")
{
if ((state == "r" ) && (node_id == 0) && (level=="AGT"))
{
pkts_0_rcvd_size[flow][inc] = pkt_size;
pkts_0_rcvd_time[flow][inc] = time;
packet_end_time[flow][packet_id] = time;
inc++;
}

if ((state == "s" ) && (pkt_type == "tcp") && (level=="RTR"))
{
packet_start_time[flow][packet_id] = time;
sender_node[flow] = node_id;
if(time < start_time[flow] ) start_time[flow] = time;
}
}
}
}
```

```
END {

for(i=1;i<=cnxs;i++)
{
if       (FILENAME == "Blackhole.tr"){file_name = "files/blackhole_"
i ".csv"}
else if (FILENAME == "DoS.tr")         {file_name = "files/dos_" i
".csv"}
else if (FILENAME == "Flood.tr")      {file_name = "files/flood_" i
".csv"}
else if (FILENAME == "Normal.tr")     {file_name = "files/normal_" i
".csv"}

sum=0.00;
k= 0;
recvnum=0;
packet_duration=0.00;
delay[i]=0;
PROCINFO["sorted_in"] = "@ind_num_asc";

for (j in pkts_0_rcvd_size[i])
{
packet_0_end_time[i][k] = pkts_0_rcvd_time[i][j];
rcvd_pkts_size[i][k]    = pkts_0_rcvd_size[i][j];
k++;
}

for ( j in packet_0_end_time[i])
{
tot_rcvd_pkts_size[i][int(j/5)] += rcvd_pkts_size[i][j];
tot_duration[i][int(j/5)] = packet_0_end_time[i][j];
}

Start_t = start_time[i];

printf("i,")              > (file_name);
printf("st,")             > (file_name);
printf("et,")             > (file_name);
printf("Average_delai,")  > (file_name);
printf("Throughput,\n")   > (file_name);

for (j in tot_duration[i])
{
for ( n in packet_end_time[i] )
{
start = packet_start_time[i][n];
end = packet_end_time[i][n];
if ((start >= Start_t) && (start < tot_duration[i][j]))
{
packet_duration = end - start;
if ( packet_duration > 0 )
{sum += packet_duration; recvnum++;}
}
}
if (recvnum != 0) delay[i]=sum/recvnum;
```

```
Delai_time = tot_duration[i][j] - Start_t;
printf("%d,", i)>> (file_name);
printf("%.4f,", Start_t)>> (file_name);
printf("%.4f,", tot_duration[i][j]) >> (file_name);
printf("%.9f,",delay[i])  >> (file_name);
printf("%.4f\n",      (tot_rcvd_pkts_size[i][j]*8)/Delai_time/1024)>>
(file_name);

Start_t = tot_duration[i][j];
sum = 0;
recvnum=0;
delay[i]=0;
}
}
}
```

Annexe C

Script test R

```
b_vp<-0
b_vn<-0
b_fp<-0
b_fn<-0
d_vp<-0
d_vn<-0
d_fp<-0
d_fn<-0
f_vp<-0
f_vn<-0
f_fp<-0
f_fn<-0
```

```
i<-1
l<-7
library(readr)

myfiles<-list.files("/home/ismail/Bureau/application/files",
pattern="*.csv",full.names = TRUE)


k<-1
j<-1
j2<-21+j

for (j in 1:28) {
  #le test blackhool
  if (j>=1 && j<=7){


    group<-1;

    dat1<-read_csv(myfiles[j])[,1:5]
    #dat1<-dat1[,-6]
    dat1<-cbind(dat1,group)
    group<-2
    dat2<-read_csv(myfiles[j2],skip_empty_rows = TRUE)[,1:5]
    #dat2<-dat2[,-6]
    dat2<-cbind(dat2,group)
    dat3<-rbind(dat1,dat2)
    ta<-dat3$Average_delai#delai_moyenne
    dl<-dat3$Throughput#débit

    model<-lm(cbind(Average_delai,Throughput) ~ group,data=dat3)
```

```
#appliquer manova
man<-manova(model)

    a<-summary(man)
    print(a)
    as.list(a)
    library (plyr)
    df <- ldply (a, data.frame)#to dataframe
    p<-df[6,14]
    print(p)
    r1= "cas attack"
    r2="cas normal"
    r3=""
    if(is.null(p)){r3="cas indeterminu"}else{
      if (p<0.05) {
        r3<-r1
        print(r3)
        b_vp<-b_vp+1


      }else{
        b_fn<-b_fn+1

        r3<-r2


      }
    }



    capture.output(print(paste("NuM=:",k)),r3,file
="/home/ismail/Bureau/application/results/resblackhole.txt",append  =
TRUE)


    k<-k+1
    j2<-j2+1
    #les tests de dos
  }else if (j>=8 && j<=14) {

    j2<-j+14
    print(myfiles[j])
    print(myfiles[j2])


    group<-1;

    dat1<-read_csv(myfiles[j])[,1:5]
    #dat1<-dat1[,-6]
    dat1<-cbind(dat1,group)
    group<-2
    dat2<-read_csv(myfiles[j2],skip_empty_rows = TRUE)[,1:5]
    #dat2<-dat2[,-6]
    dat2<-cbind(dat2,group)
    dat3<-rbind(dat1,dat2)
```

```
    ta<-dat3$Average_delai#delai moyenne
    dl<-dat3$Throughput#debit

    model<-lm(cbind(Average_delai,Throughput) ~ group,data=dat3)
    man<-manova(model)

    a<-summary(man)
    print(a)
    as.list(a)
    library (plyr)
    df <- ldply (a, data.frame)#to dataframe
    p<-df[6,14]
    print(p)
    r1="cas attack"
    r2="cas normal"
    r3=""
    if(is.null(p)){r3="cas indeterminu"}else{
      if (p<0.05) {
        r3<-r1
        print(r3)
        d_vp<-d_vp+1

      }else{
        r3<-r2
        print(r3)
        d_fn<-d_fn+1

      }
    }



    capture.output(print(paste("NuM=:",k)),r3,file
="/home/ismail/Bureau/application/results/resdos.txt",append = TRUE)


    k<-k+1
    j2<-j2+1
  }
  #cas flood

  else if (j>=15 && j<=21) {

    j2<-j+7
    group<-1;

    dat1<-read_csv(myfiles[j])[,1:5]
    #dat1<-dat1[,-6]
    dat1<-cbind(dat1,group)
    group<-2
    dat2<-read_csv(myfiles[j2],skip_empty_rows = TRUE)[,1:5]
    #dat2<-dat2[,-6]
    dat2<-cbind(dat2,group)
    dat3<-rbind(dat1,dat2)
    ta<-dat3$Average_delai#delai moyenne
    dl<-dat3$Throughput#debit
```

```
    model<-lm(cbind(Average_delai,Throughput) ~ group,data=dat3)
    man<-manova(model)

    a<-summary(man)
    print(a)
    as.list(a)
    library (plyr)
    df <- ldply (a, data.frame)#to dataframe
    p<-df[6,14]
    print(p)
    r1="cas attack"
    r2="cas normal"
    r3=""
    if(is.null(p)){r3="cas indeterminu"}else{
      if (p<0.05) {
        r3<-r1
        print(r3)
        f_vp<-f_vp+1

      }else{
        r3<-r2
        print(r3)
        f_fn<-f_fn+1

      }
    }



    capture.output(print(paste("NuM=:",k)),r3,file
="/home/ismail/Bureau/application/results/resflood.txt",append      =
TRUE)


    k<-k+1
    j2<-j2+1


  }
  #cas normal
  else{

    j2<-j
    group<-1;

    dat1<-read_csv(myfiles[j])[,1:5]
    #dat1<-dat1[,-6]
    dat1<-cbind(dat1,group)
    group<-2
    dat2<-read_csv(myfiles[j2],skip_empty_rows = TRUE)[,1:5]
    #dat2<-dat2[,-6]
    dat2<-cbind(dat2,group)
    dat3<-rbind(dat1,dat2)
    ta<-dat3$Average_delai#delai moyenne
    dl<-dat3$Throughput#debit
```

```
    model<-lm(cbind(Average_delai,Throughput) ~ group,data=dat3)
    man<-manova(model)

    a<-summary(man)
    library (plyr)
    df <- ldply (a, data.frame)#to dataframe
    p<-df[6,14]

    r1=" cas attack"
    r2="cas normal"
    r3=""
    if(is.null(p)){r3="cas indeterminu"}else{
      if (p<0.05) {
        r3<-r1
        print(r3)
        b_fp<-b_fp+1
        d_fp<-d_fp+1
        f_fp<-f_fp+1
      }else{
        r3<-r2
        print(r3)
        b_vn<-b_vn+1
        d_vn<-d_vn+1
        f_vn<-f_vn+1

      }
    }







    k<-k+1
    j2<-j2+1



  }




}

capture.output(print("Blackhole:"),print("vrai_positiv_blackhoole
:"),b_vp,print("vrai_negative_blackhool"),b_vn,print("faux_positive_
blackhole"),b_fp,print("faux_negative_blackhole"),b_fn,print("Dos:")
,print("vrai_positiv_dos
"),d_vp,print("vrai_negative_dos"),d_vn,print("faux_positive_dos"),d
_fp,print("faux_negative_dos"),d_fn,
print("FLOOd:"),print("vrai_positiv_flood :")
,f_vp,print("vrai_negative_flood"),f_vn,print("faux_positive_flood")
,f_fp,print("faux_negative_flood"),f_fn,
```

```
print("incertitude blackhole"),(b_vp+b_vn)/(b_vp+b_vn+b_fp+b_fn),
print("incertitude dos"),(d_vp+d_vn)/(d_vp+d_vn+d_fp+d_fn),
print("incertitude flood"),(f_vp+f_vn)/(f_vp+f_vn+f_fp+f_fn),
print("précision _blackhole:"),b_vp/(b_vp+b_fp),
print("précision _Dos:"),d_vp/(d_vp+d_fp),
print("précision flood:"),f_vp/(f_vp+f_fp),
print("rappel_blackhole:"),b_vp/(b_vp+b_fn),
print("rappel_dos:"),d_vp/(d_vp+d_fn),
print("rappel_flood:"),f_vp/(f_vp+f_fn),
file  ="/home/ismail/Bureau/application/results/resids.txt",append  =
TRUE)
```

Références

[1]http://www-igm.univ-mlv.fr/~dr/XPOSE2006/AurelieBunel/

[2] http://www-igm.univ-mlv.fr/~dr/XPOSE2006/AurelieBunel/Presentation.html

[3] Réseaux de capteurs, E.Fleury et David Simplot-Ryl, http://www2.lifl.fr/~simplot/NEW/

[4] Boukerche A. Werner Nelem Pazzia R., Borges Araujo R. "Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments"; Journal of Parallel and Distributed Computing, Avr 2006. - 4 : Vol. 66. - pp. 586-599

[5] https://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_15.html

[6]. https://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_34.html

[7] Akyildiz I. F. Su W., Sankarasubramaniam Y. et Cayirci E. "Wireless sensor networks: a survey", Computer Networks. Vol. 38(4). - pp. 393-422, 2002

[8] Tinyos. http:/www.tinyos.net/

[9] Contiki. web site. http ://www.sics.se/contiki.

[10] Adam Dunkels, Bj• ornGr• onvall, and Thiemo Voigt. Contiki { a lightweight and exible operating system for tiny networked sensors. IEEE EmNetS-I, November 2004. Swedish Institute of Computer Science.

[11] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, “Wireless Sensor Network Security: A Survey ”, *Department of Computer Science Wayne State University*.

[12] H. Krawczyk and M. Bellare and R. Canetti, "HMAC : Keyed-Hashing for Message Authentication", RFC 2104, 1997, February.

[13] M Healy, T Newe, and E Lewis. E_ciently securing data on a wireless sensor network.
Journal of Physics : Conference Series 76, 2007. Sensors and their Applications XIV(SENSORS07).

[14] Germano Guimar~aes, Eduardo Souto, Djamel Sadok, and Judith Kelner. Evaluation
of security mechanisms in wireless sensor networks. IEEE Proceedings of the 2005 Systems Communications (ICW'05), 2005.

[15] Germano Guimar~aes, Eduardo Souto, Djamel Sadok, and Judith Kelner. Evaluation
of security mechanisms in wireless sensor networks. IEEE Proceedings of the 2005 Systems Communications (ICW'05), 2005.
[16]https://www.researchgate.net/publication/308203983_Systeme_de_detection_d'intrusion_IDS

[17]http://lehmann.free.fr/RapportMain/node10.html

[18] https://fr.m.wikipedia.org/wiki/Fichier:Aval_NIDS.png

[19] https://fr.wikipedia.org/wiki/Systéme detection_d'intrusion

[20]http://cybercriminalite-solutions.over-blog.com

[21]S. Axelsson. Intrusion detection systems : A survey and taxonomy. Technical report, Technical report, 2000.
[22]H. Debar, M. Dacier, and A. Wespi. Towards a taxonomy of intrusion-detection systems. Computer Networks, 1999.
[23] H. Debar, M. Dacier, and A. Wespi. Towards a taxonomy of intrusion-detection systems. Computer Networks, 1999.

[24] Carl Hartung, James Balasalle, and Richard Han. Node compromise in sensor networks : The need for secure systems. Technical report, Department of Computer Science University of Colorado at Boulder, January 2005.
[25] https://www.slideshare.net/cutemanharmick/black-hole-attack-la-notion-de-trou-noir-attaque-prsent-par-dr-harmick-makiese

[26] Virgil Gligor, Bryan Parno, and Adrian Perrig. Distributed detection of node replication attacks in sensor networks. University of Maryland & Carnegie Mellon University, 2008.
[27] https://www.researchgate.net/figure/Hello-flooding-attack_fig4_259624794

[28] Hakima Chouachi and Maryline Laurent-Maknavicius. La s´ecurit´e dans les r´eseaux sans fil mobiles 3, volume 3, chapter S´ecurit´e dans les r´eseaux de capteurs sans fil, page 291. April 2007.

[29] https://fr.sawakinome.com/

[30]https://support.minitab.com/fr-fr/minitab/18/help-and-how-to/modeling-statistics/anova/supporting-topics/basics/understanding-manova/

[31] https://www.real-statistics.com/multivariate-statistics

[32] https://y-baddi.developpez.com/tutoriels/ns2/

[33] https://www.sqlpac.com/referentiel/docs/unix-awk-tutoriel.html

[34] http://chideux.e-monsite.com/pages/content/qu-est-ce-que-spss