République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Djillali Liabès de Sidi-Bel-Abbès

Faculté des Sciences Exactes

Département d'informatique

Thèse

Pour l'obtention du Diplôme de Doctorat LMD en Informatique

Option : Informatique

Présenté par :

Mr. AZZA Mohammed

Sécurité des Réseaux AD HOC

Directeur de thèse :

Pr. Kamel Mohamed Faraoun

Professeur à l'université Djillali Liabès

Devant le jury composé de :

Président :	Dr. Gafour Abdelkader	M.C.A	Université Djillali Liabès
Examinateur :	Dr. Adjoudj Reda	M.C.A	Université Djillali Liabès
Examinateur :	Dr. Keskes Nabil	M.C.A	ESI-SBA
Examinateur :	Dr. Kadri Benamar	M.C.A	Université Abou Bekr Belkaid
Examinateur :	Dr. Boukli Hacene Sofiane	M.C.A	Université Djillali Liabès
Directeur de thèse :	Pr. Faraoun Kamel Mohamed	Professeur	Université Djillali Liabès

Remerciements

Je tiens à remercier particulièrement et chaleureusement Monsieur Kamel Mohamed FARAOUN, professeur à l'université de sidi bel abbes, pour son encadrement, sa patience, sa présence, ses conseils très précieux, son dévouement et sa disponibilité à l'élaboration de cette thèse.

Je tiens à remercier chaleureusement tous les membres du jury qui ont accepté de juger ce travail, le président du jury le dr. Abdelkader GAAFOUR, ainsi les rapporteurs dr. Reda ADJOUDJ et dr.Sofiane BOUKLI HACENE ainsi les examinateurs le dr. Nabil KESKES et le dr. Benamar KADRI.

Mes remerciements vont aussi à tous les membres du laboratoire EEDIS (Evolutionary Engineering and Distributed Information Systems) pour le cadre de travail.

J'adresse aussi mes sincères remerciements à tous ceux qui ont contribué directement ou indirectement à l'aboutissement de cette thèse. Merci à tous les membres de ma famille, à mes soeurs et à mes frères, à mon père qui m'a beaucoup soutenu et encouragé et à ma mère à qui je dédie tout ce travail et qui m'a donné le courage de le mener à bien.

Enfin, ce travail n'a pu atteindre ses objectifs sans la contribution de prés ou loin de plusieurs personnes auxquelles j'adresse mes chaleureux remerciements

À mon père, ma mère,

mes frères et mes sœurs
Je vous aime !

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

AC	Access Category
ACK	Acknowledgment
AIFS	Arbitration Interframe Space
AODV	Ad hoc On demande Distance Vector
BSS	Basic Service Set
CBR	Constant Bit Rate
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detect
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
DIFS	Distributed Inter Frame Spacing
DSSS	Direct Sequence Spread Spectrum
EIFS	Extended Inter Frame Spacing
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
LLC	Logical Link Control
MAC	Meduim Access Control
MANET	Mobile Ad Hoc NETwork
MSDU	MAC Service Data Unit
MPDU	MAC Protocol Data Unit
NAV	Network Allocation Vector
NS-2	Network Simulator
P2P	Peer-to-Peer
PC	Point Coordinator
PCF	Point Coordination Function
PIFS	Priority Inter Frame Spacing
PLCP	Physical Layer Convergence Procedure
PSM	Power Saving Mechanism

Chapitre 1

Introduction générale

1.1 Introduction Générale

Le développement technologique qu'a vu le monde d'aujourd'hui à toucher tous les domaines, particulièrement le secteur de la communication qui connait une évolution considérable par l'apparition des technologies sans fil. Les communications sans fil permettent aux nœuds mobiles de transmettre leurs informations avec une grande flexibilité d'utilisation dans des zones ouvertes. En particulier, ils offrent la mise en réseau des sites dont le câblage serait trop difficile et coûteux à réaliser.

Généralement, les réseaux mobiles sans fil sont divisés en deux catégories (les réseaux avec infrastructure ou cellulaire et sans infrastructure ou ad hoc). Plusieurs systèmes utilisent le modèle cellulaire ou ce qu'on appelle les réseaux téléphoniques à commutation par circuit dont les liens de communication sont basés sur les ondes radio. Ce type nécessite une infrastructure fixe, cependant il a un grand usage dans notre vie actuelle. Tandis que la deuxième catégorie s'appelle les réseaux mobiles Ad-hoc.

Un réseau Ad-hoc est défini comme un ensemble des nœuds mobiles reliés entre eux par des liens hertziens formant des zones de transmission temporaire et dynamique. Parmi leurs caractéristiques, ces réseaux n'ont pas besoin d'une administration centralisée ou d'un médium de transmission physique. D'autre part, il n'a pas une taille fixe ou un nombre bien précis des nœuds qui peuvent joindre le réseau, de plus les nœuds sont libres, cela signifie que ces réseaux sont très adaptés dans certains domaines d'applications. Les réseaux Ad-hoc sont exploités dans plusieurs applications, on prend l'exemple le plus cité le domaine militaire qui a été le point de départ de ses réseaux. Ainsi, les applications de secours, les missions de recherche et explorations et aussi les applications médicales.

Dans un réseau Ad-hoc, généralement l'émetteur et le récepteur ne se trouvent pas dans la même zone de couverture, cela nécessite la coopération d'autres nœuds pour que l'acheminement des données se réalise. L'acheminement des données vers la bonne destination est assuré via un protocole de routage multi-sauts, pour cela plusieurs protocoles ont été conçus pour ce type de réseaux qui sont adapté par rapport à leurs caractéristiques. Ces protocoles prennent en considération les différents facteurs telle que la mobilité du nœuds, l'auto-organisation, la bande passante, le nombre des liens et les ressources rares de réseau. Pratiquement les protocoles de routage ont été classifiés en trois classes : protocoles proactifs, réactifs et hybrides selon le groupe *Mobile Ad hoc Network* (MANET) de L'IETF qui normalise les protocoles dans les réseaux ad hoc.

La catégorisation des protocoles de routage plat multi-sauts dans les réseaux mobiles ad hoc se base sur la manière de découvrir les chemins. Les protocoles

proactifs utilisent un échange périodique d'informations de routage vers tous les nœuds de la topologie pour la création et la mise à jour des routes. Les routes sont établies avant qu'il y aura une demande de transmission. Cette catégorie est basée sur deux principales techniques : le routage par vecteur de distance et le routage par état de lien. Tandis que, les protocoles réactifs établissent les chemins uniquement à la demande. La station source lance le processus de découverte de route lorsqu' elle souhaite envoyer des paquets vers un destinataire dont le chemin est inconnu. La troisième catégorie des protocoles dite hybrides, elle combine les deux techniques des protocoles proactifs et réactifs. Autrement, il existe d'autres classifications qui focalisent sur divers critères et stratégies de sélection des routes.

Le changement de la topologie et le manque de support physique provoque la collaboration des nœuds pour créer des routes vers la destination. Tous les participants dans ce processus ont le même niveau de confiance, cela donne la possibilité à quelques nœuds de changer leurs comportements principaux et deviennent malicieux.

1.2 Contributions de cette thèse

L'objectif de notre thèse entre dans le cadre de l'étude du problème de la sécurité du routage dans les réseaux mobiles ad hoc. Plusieurs travaux de recherche existent dans la littérature, et qui se font à l'heure actuelle, dans le but de sécuriser le processus de routage contre les nœuds avec des comportements malveillants. L'acheminement des données vers la bonne destination devient un défi globale dans la majorité des travaux de recherche.

Le succès tenu par les applications sans fil apparue dépend de leurs dégrées de sécurité, notamment sur leurs disponibilités et fiabilités. En outre, mettre en œuvre un service ou une architecture de sécurité nécessite une entité centralisée, cela n'existe pas dans un réseau mobile ad hoc qui devient un challenge à traiter. La communication dans un environnement mobile ad hoc est totalement libre, car elle est faite à travers des ondes radio (hertziens). Chaque nœud inclut dans une zone de transmission active peut supprimer ou modifier les informations qui circulent, par conséquent, l'intégrité des données doit être présente. Autrement, le support de transmission est ouvert, car il est basé sur la méthode CSMA ce qui facilite leur accès. Cette propriété généralement exploitée par des nœuds compromis. On prend les attaques de type déni-de-service *(DOS)* qui sont faciles à mettre en œuvre. Un nœud peut transmettre directement des paquets sans attendre son tour, inonder rapidement le réseau ce qui le rend ainsi inopérable.

Le routage est la phase essentielle de chaque transmission de donnée, cependant, c'est un grand problème d'avoir un routage fiable dans les réseaux mobiles ad hoc suite aux manque d'une architecture fixe et d'une administration centralisée. De plus les protocoles de routage multi-sauts établissent les routes avec la coopération des voisins pour atteindre la destination, donc la relation entre les paires est considérée confiante et sûre. Cela permet aux nœuds malveillants de détourner les communications à sa destination réelle sans que la station source n'en prenne conscience.

Les attaques contre les protocoles de routage sont nombreux, d'habitude elles sont classifiés en deux classes : les attaques actives et passives. Dans une attaque active le nœud malveillant détourne le processus de routage et causant ainsi de graves dégats dans les réseaux. L'attaque *black hole* est une attaque très dangereuse, où le nœud malicieux absorbe tous les paquets de données vers lui, ensuite il les supprime sans la connaissance du station destination.

Des solutions ont été proposées pour détecter et isoler les nœuds malveillants. Ces solutions assurent que les décisions de routage d'un nœud doivent être contrôlées. Plusieurs techniques et mesures de sécurité ont été introduites, nous pouvons les classés dans cinq catégories générales : Technique de détection basée sur le numéro de séquence, sur la cryptographie, sur le point de vue des nœuds voisins, sur les systèmes de détections d'intrusion (IDS), et d'autres basée sur la confiance.

Le travail de cette thèse consiste à définir des mécanismes de sécurité adaptés au protocole de routage (AODV) dans les réseaux ad hoc mobiles, on prend en considération l'attaque de suppression des paquets données. Les attaques menées par des nœuds malicieux, telle que *black hole* peuvent perturber la disponibilité de service. Notre première contribution garantie la détection et l'isolation du nœud *black hole*. Nous traitons la sécurité à travers une architecture de communication inter-couche entre la couche MAC et réseaux.

La majorité des travaux existants traitent chaque attaque de manière absolue. D'autres travaux qui se basent sur la réputation du nœuds voisins et permettant de détecter les nœuds qui ont un comportement malveillant ou égoïste. La deuxième proposition est une méthode améliorée, elle se base sur la réputation direct des nœuds voisins. Mais souvent, il est très difficile de proposer des solutions assez fiables et robustes devant les différentes attaques existantes, sans affecter les performances des protocoles de routage du réseau ad hoc. Notamment, un protocole de routage ad hoc doit s'appuyer sur une architecture globale de confiance et doit introduire des mécanismes contre la majorité des attaques actives.

1.3 Organisation de la thèse

Ce manuscrit est organisé en cinq chapitres suivis d'une conclusion générale.

Une représentation globale sur les éléments qui constituent l'environnement de notre travail est présentée dans les trois premiers chapitres, après nos contributions sont détaillées dans les deux derniers.

Nous présenterons dans le deuxième chapitre un aperçu général sur les réseaux ad hoc, leurs concepts, leurs caractéristiques, ainsi que leurs applications. Après, Nous détaillons le rôle des deux couches inférieures du modèle OSI, et plus précisément la fonction de coordination distribuée (DCF) pour accéder au support de la couche MAC.

Le troisième chapitre se focalise sur les protocoles de routage. Au début, nous parlerons sur leurs fonctionnements, les contraintes auxquelles ils doivent faire face. En suite, nous présenterons une classification de ces protocoles. Nous décrivons quelques protocoles de routage, les plus élaborés et développés pour les réseaux ad hoc, après ça, nous détaillons le protocole AODV utilisé dans cette thèse. En suite, Nous expliquons quelques attaques et leurs conséquences sur les performances au niveau de la couche de routage. A la fin, nous aborderons la problématique de la sécurité dans les réseaux ad hoc en expliquant les différents types d'attaques qui peuvent viser le réseau ainsi que les outils de base de sécurité existants et qui sont développés pour faire face à ces menaces.

Le quatrième chapitre sera dédié à notre première contribution qui s'intitule la communication inter-couches pour détecter et éliminer les nœuds malveillants qui exploitent l'attaque black hole dans le protocole AODV. Nous présentons en premier lieu en détail le principe général de notre approche suivi par une étude des performances de notre approche en effectuant différents tests, ensuite, nous interprétons les résultats obtenus.

Le chapitre cinq aborde notre deuxième proposition basée sur la réputation directe, Nous commencerons d'abord par présenter les motivations de cette proposition (les techniques basées sur l'écoute et sur l'acquittement). Après, nous décrivons notre amélioration qui consiste à introduire les paquets de données non transmis à cause des circonstances du réseau mobile ad hoc comme l'épuisement d'énergie, aussi la surcharge de la file d'attente et la mobilité du nœud voisin. Enfin un ensemble des simulations ont été faite, sous différents scénarios qui ont été analysés. Les résultats obtenus montrent l'efficacité de l'amélioration qui a été réalisée.

Le dernier chapitre est consacré à la conclusion et les perspective sur la sécurité du protocole de routage sous des comportements malveillants.

Chapitre 2

Généralités sur Les Réseaux Ad-hoc

Contents

2.1 Introduction

Les réseaux mobiles Ad-hoc (MANET) forment un nouveau paradigme des réseaux sans fil. Ils se constituent par l'interconnexion de différentes entités mobiles inconnues et ne reposent sur aucune infrastructure fixe ou contrôle centralisé. La coopération entre ces entités permet de maintenir les services du réseau. La principale fonctionnalité des réseaux Ad-hoc est l'opération de routage. Elle contrôle et gère le trafic des messages dans le réseau. L'objectif principal d'un protocole de routage pour un réseau Ad-hoc est l'établissement efficace d'itinéraires entre une paire de nœuds de telle sorte que les messages puissent être acheminés. Le protocole de routage permet aux nœuds de se connecter directement les uns aux autres pour relayer les messages par des sauts multiples.

Dans cette partie, nous présenterons dans la section 2.2 les détails des réseaux Ad-hoc, leurs caractéristiques, et leurs domaines d'application. Par suite, dans la section 2.3 nous présenterons les techniques utilisées par la IEEE 802.11 dans les deux premières couches du modèle OSI. Enfin, la section 2.4 conclut le chapitre.

2.2 Généralité sur Les réseaux Ad hoc

2.2.1 Définition d'un réseau ad-hoc

un réseau Ad-hoc mobile (MANET : Mobile Ad-hoc NETwork) [Johansson et al., 1999] est défini comme étant un système autonome dynamique composé de nœuds mobiles (unités mobiles) interconnecté via des connexions sans fil sans l'utilisation d'une infrastructure fixe de type point d'accès par exemple et sans administration centralisée.

Une définition de ces réseaux est donnée formellement dans RFC 2501[Corson and Macker, 1999] : "Un réseau Ad-hoc comprend des plates-formes mobiles (par exemple, un routeur interconnectant différents hôtes et équipements sans fil) appelées nœuds qui sont libres de se déplacer sans contrainte. Un réseau Ad-hoc est donc un système autonome de nœuds mobiles. Ce système peut fonctionner d'une manière isolée ou s'interfacer à des réseaux fixes à travers des passerelles".

Le modèle de réseau Ad-Hoc ne comporte pas l'entité "site fixe". Tous les sites du réseau sont mobiles et communiquent d'une manière directe en utilisant leurs interfaces de communication sans fil. L'absence de l'infrastructure ou d'un réseau filaire composé des stations de base, oblige les unités mobiles (UM) à se comporter

comme des routeurs qui participent à la découverte et la maintenance des chemins pour les autres hôtes dans le réseau.

La communication entre des nœuds dans le réseau ce fait directement, à la même manière d'un réseau *Peer-to-Peer* P2P. Les nœuds sont donc libres de se déplacer aléatoirement et s'organisent arbitrairement. Cependant, la route entre un nœud source et un nœud destination peut impliquer plusieurs sauts sans fil, d'où l'appellation des réseaux sans fil multi-sauts. Un nœud mobile peut donc communiquer directement avec un autre nœud s'il est dans sa portée de transmission. Au delà de cette portée, les nœuds intermédiaires jouent le rôle de routeurs (relayeurs) pour relayer les messages saut par saut.

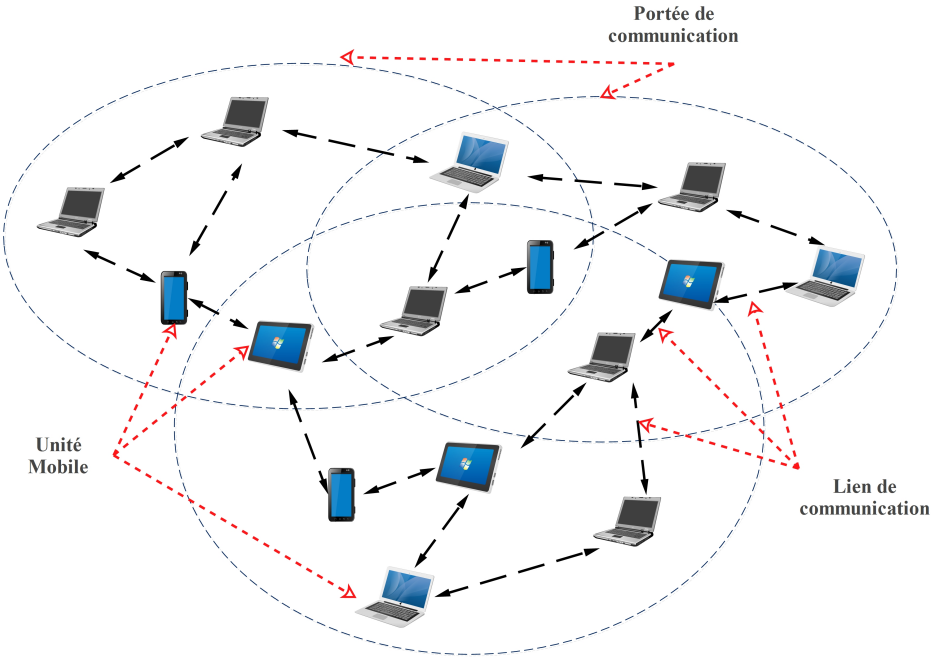

Figure 2.1 – Exemple d'un réseau Ad-Hoc

2.2.2 Modélisation d'un réseau ad-hoc

Un réseau ad-hoc peut être modélisé par un graphe $Gt = (V_t, E_t)$. Où V_t, représente l'ensemble des nœuds (i.e. les unités ou les hôtes mobiles) du réseau et E_t modélise l'ensemble des connections qui existent entre ces nœuds. Si $e = (A, B) \in E_t$, cela veut dire que les nœuds A et B sont en mesure de communiquer directement à l'instant t. La figure 2.1 donne un exemple d'un réseau ad-hoc présenté sous la forme d'un graphe dans la Figure 2.2

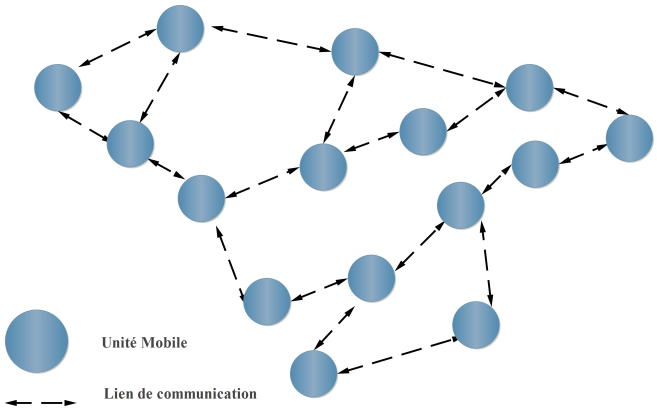

Figure 2.2 – Modélisation d'un réseau ad-hoc

2.2.3 Caractéristique du réseau ad-hoc

Les réseaux Ad-Hoc héritent les mêmes propriétés et problèmes liés aux réseaux sans fil. Le canal radio est limité en termes de capacité et débit, ce qui le rend plus exposé aux pertes (par rapport au médium filaire). Le canal est confronté aux problèmes des stations cachées et stations exposées. En outre, les liens sans fil sont asymétriques et non sécurisés.

D'autres caractéristiques spécifiques aux réseaux Ad-Hoc conduisent à ajouter une complexité et des contraintes supplémentaires qui doivent être prises en compte lors de la conception des algorithmes et des protocoles réseaux, à savoir : Les réseaux sans fil Ad-Hoc se caractérisent principalement par [Marina and Das, 2001] :

Absence d'infrastructure centralisée

Les réseaux Ad-Hoc se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue. Chaque nœud travaille dans un environnement pair à pair distribué, et agit en tant que routeur pour relayer des communications, ou générer ses propres données.

Topologie dynamique et maintenance des routes

Les nœuds sont libres de se déplacer et ce fait que la topologie du réseau est typiquement multi-sauts qui peut changer aléatoirement et rapidement n'importe quel moment. Par exemple, un nœud peut joindre un réseau, changer de position ou quitter le réseau. Ce déplacement a naturellement un impact sur la morphologie du réseau et peut modifier le comportement du canal de communication. La perte fréquente des routes entraine la dégradation des flux transmis et elle cause non seulement de grandes pertes des paquets mais aussi un délai long pour reconstituer une nouvelle route, la figure 2.3 montre le changement de topologie dans un réseau Ad-Hoc.

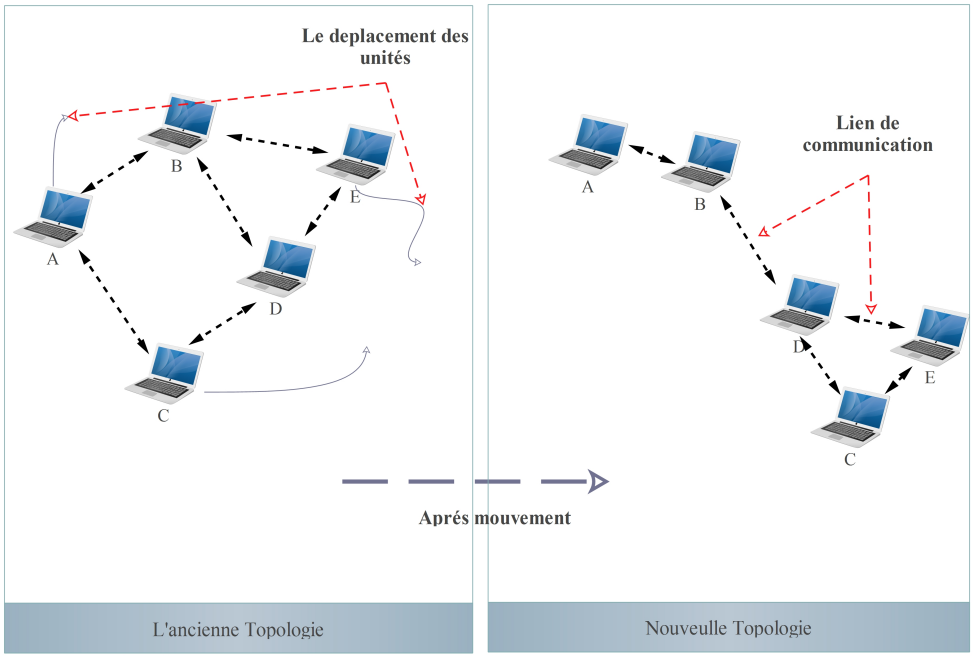

Figure 2.3 – changement de topologie

La contrainte d'énergie

Les équipements mobiles disposent de batteries ayant une charge limitée, telles que les tablettes, et par conséquent d'une durée de traitement réduite. Sachant qu'une partie de l'énergie est déjà consommée par la fonctionnalité du routage. Cela limite les services et les applications supportées par chaque nœud.

Bande passante limitée

Une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé (ondes radio). Ce

partage fait que la bande passante réservée à un hôte soit modeste.

Un débit plus faible

Par rapport à son équivalent filaire. le débit reste toujours faible par exemple le débit de la norme IEEE 802.11ac peut aller jusqu'à 433 Mbit/s

Les interférences

Les liens radios ne sont pas isolés et le nombre des canaux disponibles est limité. Il faut donc les partager. Les interférences peuvent être de natures diverses .Par exemple, des émetteurs travaillant à des fréquences trop proches peuvent interférer entre eux. L'environnement lui-même peut également produire des bruits parasites (certains équipements électriques, certains moteurs, ...) qui interférent avec les communications. L'environnement peut aussi déformer le signal et le rendre rapidement incompréhensible à cause des phénomènes d'atténuation, de réflexion ou des chemins multiples (l'atténuation et la réflexion varient en fonction des matériaux rencontrés). Le problème des chemins multiples apparait lors des réflexions d'une même onde par des chemins différents arrivent de manière décalée dans le temps au récepteur. Ces problèmes font que les taux d'erreurs de transmission dans les réseaux radio sont nettement plus élevés que dans les réseaux filaires.

L'hétérogénéité des nœuds

Un nœud mobile peut être équipé d'une ou plusieurs interfaces radio ayant des capacités de transmission variables et opérant dans des plages de fréquences différentes. Cette hétérogénéité de capacité peut engendrer des liens asymétriques dans le réseau. De plus, les nœuds peuvent avoir des différences en terme de capacité de traitement (CPU, mémoire), de logiciel, de taille (petit, grand) et de mobilité (lent, rapide). Dans ce cas, une adaptation dynamique des protocoles s'avère nécessaire pour supporter de telles situations.

Une atténuation rapide du signal

L'atténuation est en fonction de la distance (bien plus rapide que sur un câble) qui induit l'impossibilité pour un émetteur de détecter une collision au moment même où il transmet. Dans un réseau filaire, un émetteur sait qu'il y a une collision quand le signal qu'il lit sur le câble est différent de celui qu'il cherche à émettre. Dans un réseau radio, un signal venant d'un autre nœud est tellement atténué par la distance qu'il ne provoquera que des perturbations négligeables par rapport au

signal émis localement. Par exemple, sur la figure, au niveau du nœud B, le signal émis par B lui-même plus fort que celui qu'il reçoit du nœud A. Par conséquent, le signal du nœud A est complètement ignoré par B qui croit qu'il n'y a pas de collision. Malheureusement, au niveau du nœud C, les deux signaux ont des puissances comparables et il y a bien une collision du point de vue du récepteur. Seul un système d'acquittement peut garantir la bonne réception d'un message dans ce type de contexte.

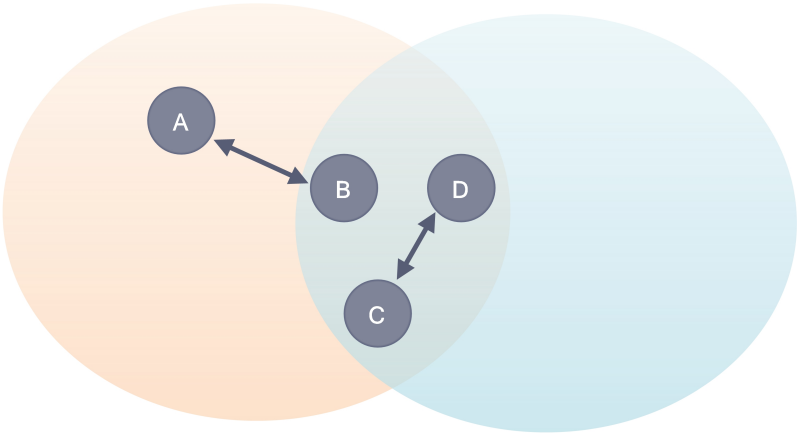

Figure 2.4 – Une atténuation rapide du signal

La puissance du signal

Non seulement elle est rapidement atténuée avec la distance, mais elle est également limitée par des réglementations très strictes. Un émetteur ne peut donc dépasser une certaine puissance à l'émission.

Le problème du nœud caché

Ce phénomène est très particulier à l'environnement sans fil. Un exemple est illustré par la figure 2.5a. Dans cet exemple, les nœuds A et B ne s'entendent pas, à cause d'un obstacle qui empêche la propagation des ondes. Les mécanismes d'accès au canal vont permettre alors à ces nœuds de commencer leurs émissions simultanément. Ce qui provoque des collisions au niveau du nœud C.

Problème de la station exposée

Ce problème est l'inverse du précédent, il survient lorsqu'une station désire établir une transmission avec une autre station et elle doit la retarder car elle détecte une transmission en cours entre deux autres stations se trouvant dans son voisinage.

Dans ce cas, seule la zone située entre B et C est sujette à des perturbations, les deux transmissions auraient donc pu prendre place simultanément voir la figure 2.5b.

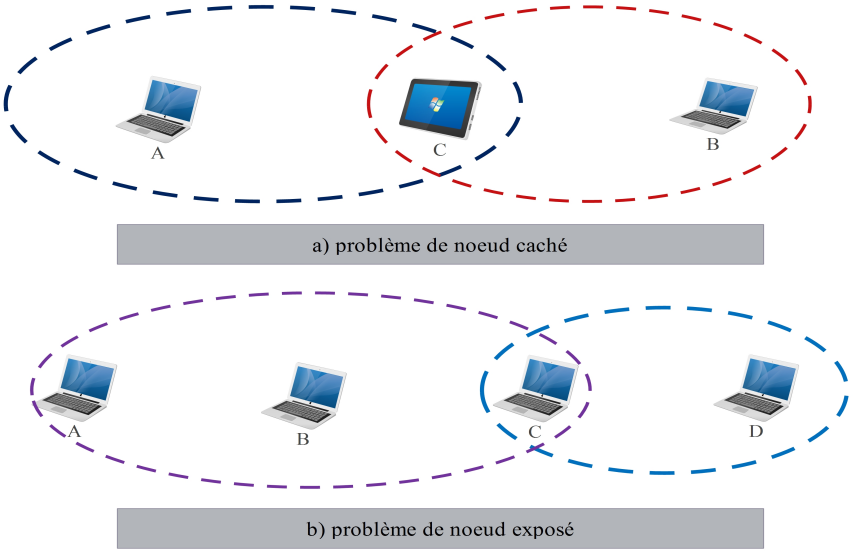

Figure 2.5 – Problème de nœud caché et nœud exposé

Une faible sécurité

Il est facile d'espionner un canal radio de manière passive. Les protections ne pouvant pas se faire de manière physique (il est en général difficile d'empêcher quelqu'un de placer discrètement une antenne réceptrice très sensible dans le voisinage), elles devront être mises en place de manière logique, en utilisant la cryptographie ou éventuellement des antennes directionnelles. Mais le canal radio restera quoi qu'il en soit vulnérable à un brouillage massif DOS (denial of service attack).

2.2.4 Avantage des réseaux ad-hoc

Les avantages les plus importants des réseaux ad-hoc sont [Hajami, 2011] :

1. Déploiement facile, rapide et économique : dans les réseaux ad-hoc, la tâche assommant de déploiement des stations de base (cablage, installation, etc.) n'est plus nécessaire. En conséquence, le déploiement est aussi plus rapide et se fait avec un faible coût.
2. Tolérance aux pannes : un réseau ad-hoc continue à fonctionner même si quelques nœuds tombent en panne, ceci est dû au fait qu'il ne comporte pas de nœuds centraux.

2.3 Architecture en couche du réseau IEEE 802.11

2.3.1 La couche physique

la couche physique de la norme IEEE 802.11 [DCF, 2004] c'est une interface située entre la couche MAC et le support, elle permet d'envoyer et de recevoir des trames de donnée. La couche physique se compose de deux sous couches (voir figure 2.6) :

1. PLCP (Physical Layer Convergence Procedure) : elle est liée directement à la couche MAC, elle permet d'écouter le support et notifier son état s'il est libre à la couche supérieure MAC.
2. PMD (Physical Media Dependent) : elle est utilisée pour l'encodage des bits et la modulation. Elle permet la transmission et la réception des données par l'intermédiaire du support sans fil. Il existe trois techniques de transmission [Bouatay, 2010] :

   Le FHSS (Frequency Hopping Spread Spectrum) Il était utilisé pour des raisons militaires pour empêcher l'écoute des transmissions. En effet, cette méthode permet de faire des sauts de fréquences sur 79 sous canaux de largeur 1 MHz durant la transmission entre l'émetteur et le récepteur. Ces derniers se mettent d'accord sur une séquence de sauts précise. cette méthode est vulnérable au attaques à cause de la séquence de saut qui est fixe à la majorité des nœuds mais elle n'empêche pas que elle diminue les interférence entre les nœuds dans une même cellule.

   Le DSSS (Direct Sequence Spread Spectrum) cette technique consiste à diviser la bande de 83.5 MHz en 14 canaux de 20 MHz de largeur. Elle est plus sensible que la première aux interférences. La répartition du point d'accès *APs* et l'affectation organisée des canaux sont recommandées pour réduire les perturbations des transmissions.

   L'infrarouge (IR) consiste a diffusé d'une lumière infrarouge d'onde sur une longueur comprise entre 850 et 950 nm.

   OFDM (Orthogonal Frequency Division Multiplexing) est une technique qui fait appel au multiplexage par la répartition des fréquences sur des porteuses orthogonales. Cette orthogonalité permet de séparer les canaux afin

d'éviter les interférences du canal. Son principe est de partitionner la bande passante en plusieurs sous porteuses ou canaux distincts. Cette distinction est assurée par la propriété d'orthogonalité où l'amplitude maximale d'une porteuse correspond à une amplitude nulle du porteuses des voisins.

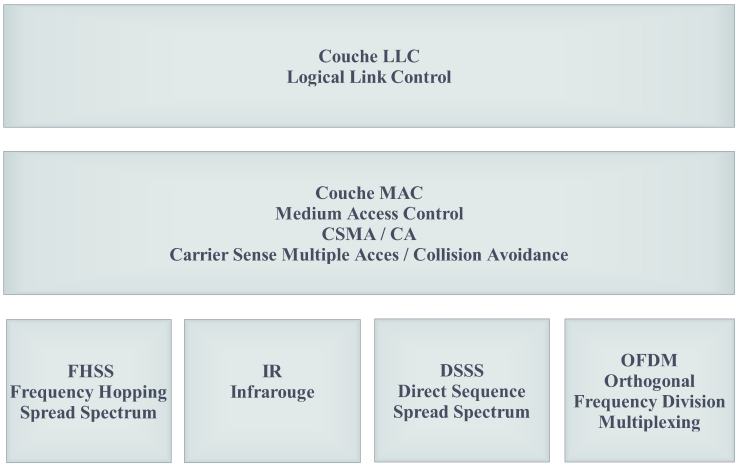

Figure 2.6 – Architecture de la couche physique et MAC [Bouatay, 2010]

2.3.2 La couche Liaison de donnée

La couche MAC est spécifiée dans la norme IEEE 802.11 [Shakkottai et al., 2003] avec une variété de fonction qui prend en charge l'opération d'accès au support sans fil. Elle gère et maintient la communication entre les stations en coordonnant l'accès à un canal radio commun, ainsi l'utilisation des protocoles qui améliorent la communication.

Le protocole IEEE 802.11 prend en charge deux types de fonctionnement d'accès : La méthode d'accès de base PCF (Point Coordination Function) et DCF (Distributed Coordination Function) [Giuseppe and Ilenia, 2005].

1. Distributed Coordination Function (DCF) : Cette méthode d'accès, assez similaire à celle d'Ethernet, est dite période de contention. Elle est conçue pour supporter les transmissions de données asynchrones tout en permettant à tous les utilisateurs d'accéder au support.
2. Point Coordination Function (PCF) : Par contre cette méthode, est dite période sans contention et ne génère pas de collision du fait que le système de transmission de données est centralisé. Ce mode est utilisé dans les réseaux avec infrastructure car il faut un point d'accès qui gère le trafic synchrone par exemple les applications à temps réel.

Fonction de Coordination Distribué(DCF)

Ce mode [Giuseppe and Ilenia, 2005] gère le trafic asynchrone à base d'un mécanisme d'accès multiple avec un évitement de collision (CSMA/CA) [Kaixin et al., 2002; Yang and Rosdahl, 2002]. Dans son principe de fonctionnement, il combine le protocole CSMA/CA avec l'algorithme back-off. Le protocole (CSMA/CA) utilise un mécanisme d'évitement de collision basé sur le principe de l'écoute au canal, l'émetteur doit s'assurer que le support est inactif avant de transmettre les données, cela signifié que la probabilité d'avoir une collision est petite.

CSMA/CA utilise des espacements d'inactivités entre les trames, ce sont des périodes de temps appelées *l'inter-Frame Spacing* (IFS). Il existe quatre types d'IFS :

1. Short Inter-Frame Spacing (SIFS) : le plus court des IFS. Il est utilisé pour séparer les différentes trames transmises au sein d'un même dialogue comme par exemple, entre des données et leurs acquittements ou entre différents fragments d'une même trame ou pour toute autre transmission relative à un même dialogue (question-réponse).
2. DCF Inter-Frame Spacing (DIFS) : est le temps que doivent attendre les autres stations avant d'émettre un paquet en mode DCF. La valeur du DIFS est égale à celle d'un SIFS augmentée de deux timeslots
3. PCF Inter-Frame Spacing (PIFS) : est le temps que doivent attendre les stations avant d'émettre un paquet en mode PCF. La valeur est inférieure au DIFS, pour favoriser ce mode.
4. Extended Inter-Frame Spacing (EIFS) : est le plus long des IFS. Lorsqu'une station reçoit une trame erronée, elle doit attendre pendant un EIFS l'acquittement de cette trame.

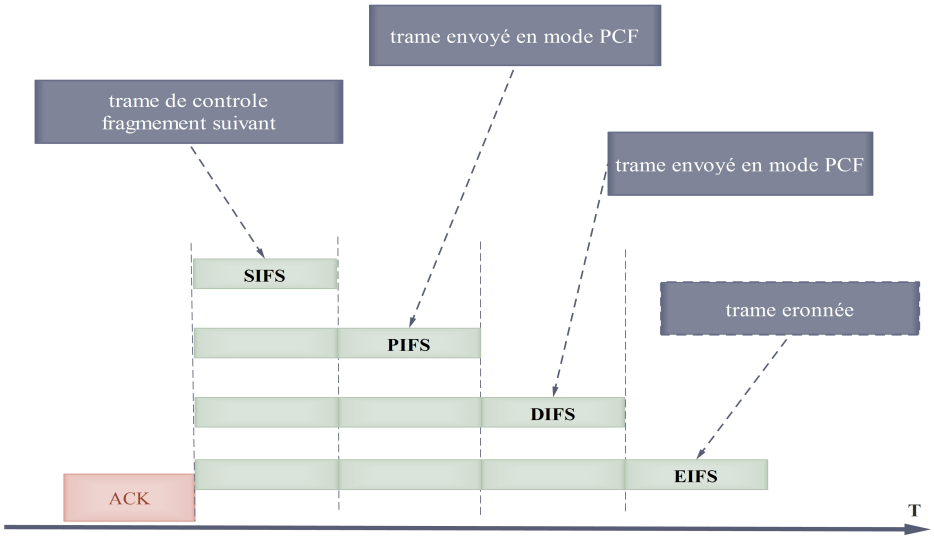

Figure 2.7 – Espacement entre trames

Si une station veut émettre au préalable elle doit écouter le canal s'il est libre ou occupé. Si le canal est occupé, la transmission est différée. Dans le cas contraire, si le média est libre pendant un temps donné DIFS, alors la station a le droit d'émettre. Suite à l' épuisement de temps calculé par l'algorithme de back-off à ce moment la station commence la transmission des données. Le calcul du temporisateur se fait par le biais de l'algorithme du back-off. Il est utilisé de la même manière que dans le CSMA/CD La seule chose qui change, c'est qu'on ne détecte pas la collision, mais, on déduit qu'il s'est produit une collision lorsqu'on ne reçoit pas d'ACK [DCF, 2004]. Il multiplie une valeur tiré aléatoirement dans un intervalle $[0, CW]$ (Contention Window) par une tranche de temps (timeslot). Ce temporisateur sera décrémenté seulement si le canal est libre. Si une collision est apparue ou une nouvelle tentative de transmission démarre.La valeur CW croit exponentiellement jusqu'à atteindre une valeur maximale, dans ce cas la transmission est échouée.

Le mode DCF est basé sur Le mécanisme de réservation de canal RTS/CTS, l'émetteur envoie une trame de contrôle RTS (Read To Send) à la station destinatrice. Tous les nœuds qui ont reçu ce RTS savent qu'une communication va voir lieu. La durée de communication est précisée dans le paquet RTS, ainsi que la station source et destination. A ce point, les voisins vont s'empêcher d'émettre pendant toute cette période définie par la trame RTS. Cette opération est réalisée grâce au NAV (Network Allocation Vector) qui stocke la valeur de cette durée et qui joue le rôle d'horloge.

Le récepteur qui reçoit le RTS renvoie la trame de contrôle CTS s'il n'est pas

lui-même bloqué par son NAV. La trame CTS (Clear To Send) a le même effet que la trame RTS pour les stations qui sont dans la même portée de communication du récepteur. À la réception du CTS, l'émetteur sait que le médium a été réservé et qu'il peut donc émettre ses données.

Après réception de toutes les données émises par la source, le récepteur envoie un accusé de réception (ACK). Toutes les stations voisines patientent pendant un intervalle temps qui est nécessaire avant la transmission du volume d'information. Cette technique permet d'éviter au maximum les collisions en laissant, pour chaque station, la même probabilité d'accès au support.

Le mode DFC peut être présenté comme le diagramme suivant :

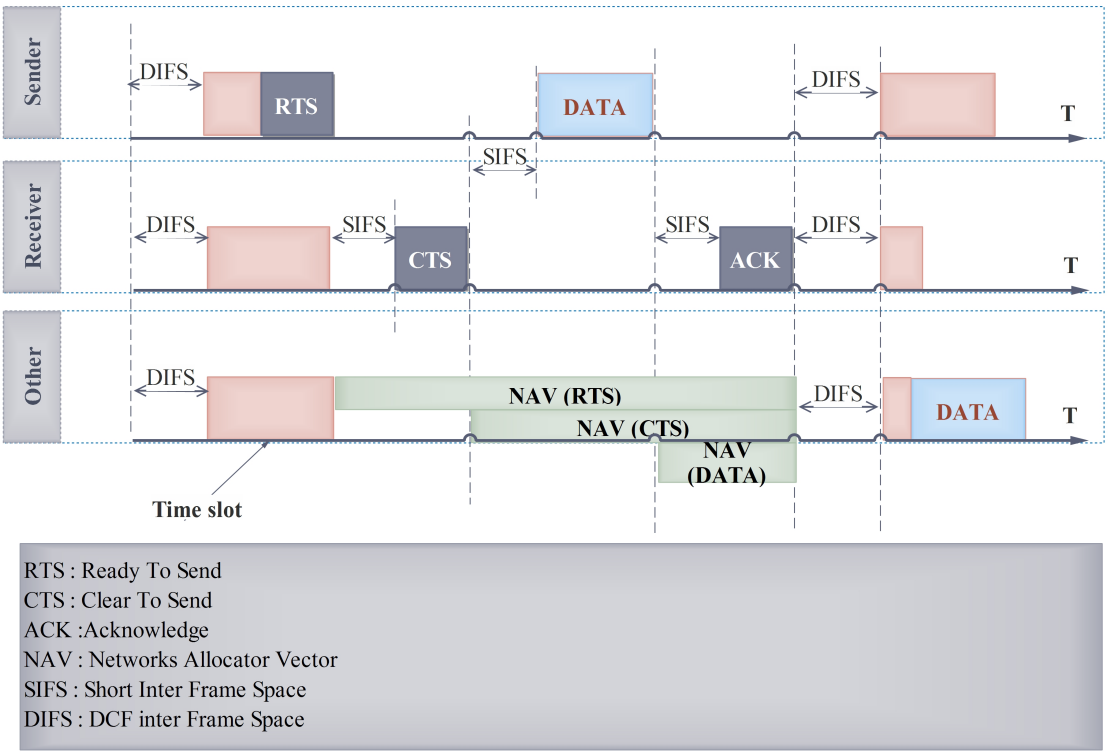

Figure 2.8 – Méthode d'accès CSMA/CA

Lors de l'utilisation du mécanisme de détection virtuelle, les collisions se produisent au niveau du RTS. La figure 2.9 illustre le comportement du protocole CSMA/CA utilisant le mécanisme RTS/CTS lorsque le RTS n'est pas reçu. Le RTS sera retransmis à cause des collisions ou des pertes.

Algorithme de backoff exponentiel BEB (Binary Exponenttiel Backoff)

Le protocole d'accès CSMA/CA est couplé avec l'algorithme de Backoff qui est

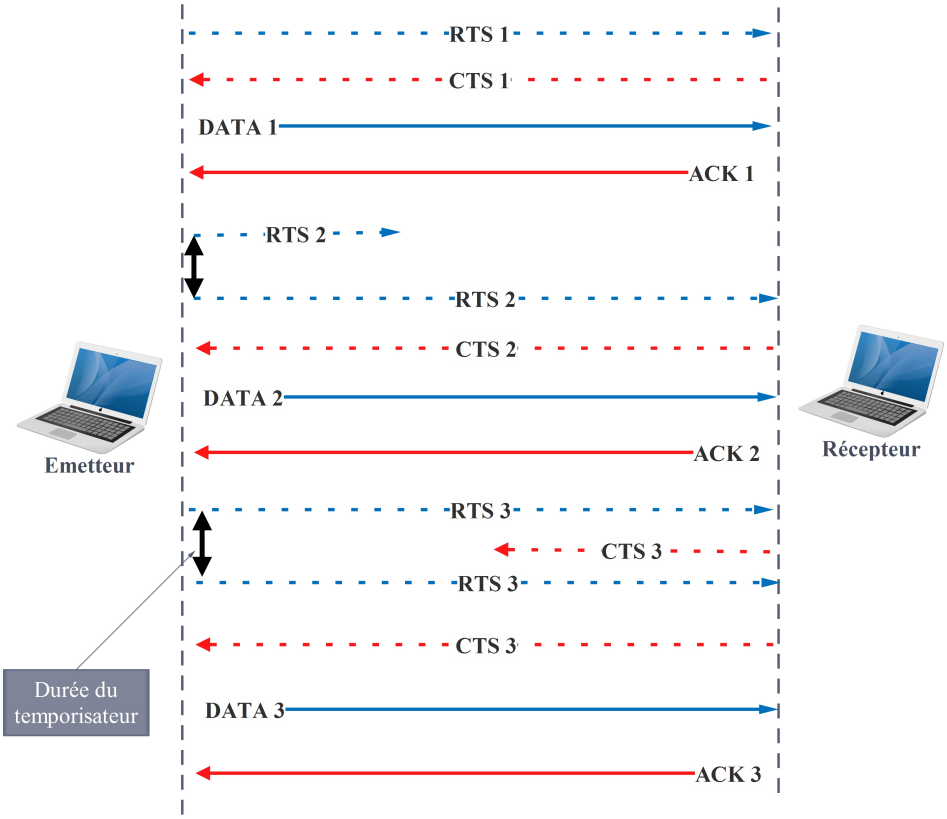

Figure 2.9 – Méthode d'accès CSMA/CA avec détection virtuelle

un mécanisme qui calcule une valeur d'attente pour gérer les transmissions et les retransmissions.

L'algorithme de Backoff est une méthode qui permet de résoudre l'accès simultané entre plusieurs stations au support. Il utilise la notion de fenêtre de contention CW (*Contention Window*). Le CW correspond au nombre maximum pour la sélection aléatoire. CW prend ses valeurs entre CW_{min} et CW_{max}.

L'algorithme de Backoff exponentiel *BEB* doit être exécuté dans les cas suivants :

- avant la première transmission d'un paquet et que le support est occupé (voir figure 2.10).
- Après chaque retransmission.
- Après une transmission réussie.

Comme exemple dans le cas du 802.11a et 802.11g. La valeur de CW_{max} égale à 1023 et à CW_{min} égale à 15.

La procédure de Backoff peut être déclenchée pour une première transmission. Cela veut dire que la station voulant émettre trouve le canal occupé pour une période d'écoute DIFS. Dans ce cas, la procédure suivante est exécutée :

- Initialisation de CW avec CW_{min}.

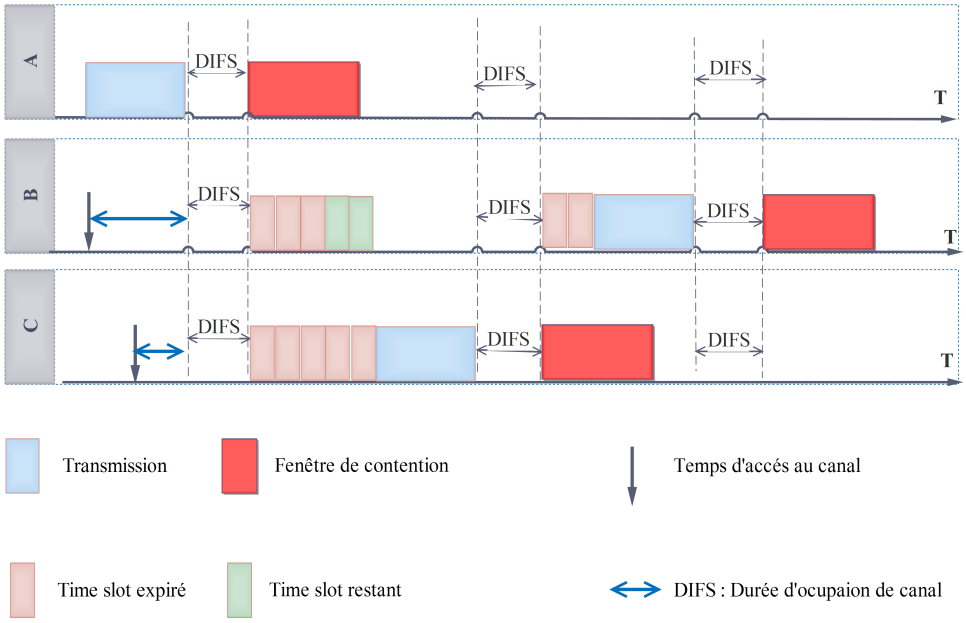

FIGURE 2.10 – Illustration de l'algorithme de Backoff

- Si le canal devient libre, après un DIFS la station commence à décrémenter son temporisateur par Time Slot.
- Si le *Backoff_Timer* est égal à 0, la station commence l'émission. Si au cours de la décrémentation, une autre station émet, la station en question bloque son $Backoff\_Timer$ et ne pourra le décrémenter que si la station finie d'émettre (canal libre avec une attente de la durée DIFS).

Lorsqu'il se produit une collision ou une perte d'acquittement au niveau de la liaison, l'algorithme de Backoff consiste à augmenter exponentiellement CW selon la formule suivante :

$$CW_{new} = 2 \times (CW_{old} + 1) - 1 \tag{2.1}$$

L'augmentation de CW s'arrête si elle atteigne la valeur CW_{max}. Après ce fait, CW reste égale à cette valeur jusqu'à ce qu'elle soit réinitialisée. En effet, CW reprend la valeur CW_{min} après une transmission réussite ou après avoir atteint la limite maximale. Le compteur du Backoff est liée à la taille de CW comme le démontre l'équation suivante :

$$Backoff\_Time = \alpha \times (0, min(CW_{min} \times 2^{\beta}, CW_{max})) \times slot\_time \tag{2.2}$$

Ou : α : génère un entier aléatoire. β : nombre de tentative de retransmission. slot_time : laps de temps spécifié dans l'unité MSDU.

2.4 Conclusion

Les réseaux mobiles Ad-hoc (ou MANET pour Mobile Ad hoc Netwokrs) sont des réseaux sans fil mobiles indépendants de toute infrastructure fixe. Vu la nature du média de transport, ces réseaux héritent des avantages et des inconvénients de leur homologue qui se rattache à des infrastructures filaires, auxquels se rajoutent des nouveaux avantages et inconvénients.

Les réseaux Ad-hoc se déroulent dans des environnements hostiles et dans des scénarios ou le câblage serait difficile. De plus, à cause de plusieurs contraintes telles que l'absence d'infrastructure, l'absence d'une relation de confiance préalable, les contraintes des ressources des nœuds, la mobilité de ces réseaux ont de nombreuses utilités.

La norme IEEE 802.11 définit un ensemble des fonctionnalités nécessaire pour accéder au support, ainsi que la méthode *CSMA/CA* qui permet d'éviter les collision. La fonction *DCF* assure la coordination entre l'émetteur et le récepteur, cette fonction utilise le mécanise *RTS/CTS* qui garantit la transmission entre les nœuds et aussi le problème de nœud caché et exposé.

Chapitre 3

Routage et la Sécurité des réseaux Ad-hoc

Contents

3.1 Introduction

Comme nous avons déjà vu dans le chapitre précédent, un réseau Ad-hoc est un ensemble de nœuds mobiles constituant une topologie dynamique sans infrastructure et une administration centralisée, où la rupture des liens de communication entre les nœuds s'apparaitre à tout moment. Dans la plupart des cas, Le nœud destination ne se trouve pas obligatoirement dans la zone de couverture du nœud source cela veut dire que le transfert de donnée se fait par des nœuds intermédiaires qui relie entre la source et la destination (communication multi-sauts). Suite aux lien de connectivité radio et aux mobilités, la position de chaque nœud n'est pas connue ainsi qu'il n'existe pas une station fixe qui contrôle ce mouvement.

Pour assurer l'acheminement des données en établissant une architecture globale qui prend en compte les différentes caractéristiques du réseau Ad-hoc. Ce type de réseau a des particularités qui sont complètement différentes par rapport aux réseaux filaires, donc les protocoles de routage sont conçus pour établir des routes fiables entre les nœuds. Les protocoles de routage peuvent être classifier suite à plusieurs techniques et stratégies de découverte et maintenance du route. Généralement il existe trois catégories principales selon le groupe MANET de L'IETF *L'Internet Engineering Task Force* qui normalise les protocoles dans le réseau ad hoc qui sont : les protocoles proactifs, réactifs et hybrides.

Le souci de sécurité dans les opérations de routage représente un défi principal dans la conception de ces protocoles. Suite à, l'absence d'une entité centrale ou d'une infrastructure fixe, les solutions de sécurité classiques ne sont pas adaptées au contexte des réseaux Ad-hoc. Les vulnérabilités dans ces réseaux sont nombreuses : usurpation d'identité, nœuds égoïstes ou malveillants, fabrication, modification ou suppression de trafic réseau, etc.... D'autant plus que chaque nœud dans le réseau représente un point de vulnérabilité, car chaque élément contribue par un rôle important dans le bon fonctionnement général du processus de routage. En particulier, si aucun mécanisme n'est mis en place pour permettre à chaque nœud de déterminer le bon fonctionnement et de vérifier la cohérence des données de routage, le nœud accepte les informations de routage venant d'autre nœud du réseau. Par conséquent, un attaquant peut envoyer des messages déclarant des informations incorrectes sur le réseau, afin d'y mener ensuite des actions malveillantes.

Dans ce chapitre, dans la section 3.2 nous définirons le routage ainsi que les caractéristiques qui empêchent le bon déroulement du protocole de routage. Ensuite, nous décrirons dans la section 3.4 chaque catégorie du protocole de routage (Proactif, Réactif, Hybride) dans les réseaux Ad-hoc avec leur spécificités et particularités en expliquant dans chaque cas, quelques exemples. Dans la section 3.5 nous détaillerons

le protocole AODV, ensuite nous donnerons un aperçu général sur ses principes de découverte et maintenance de route et les paquets de contrôle utilisé. Ensuite, nous détaillerons dans la section 3.6 toutes les aspects de sécurité en terme des objectifs et en terme des vulnérabilités qui menacent ce type de réseau. Dans la section 3.7, nous décrirons les principaux techniques et outils cryptographiques existant dans le domaine de sécurité. Quelques attaques au niveau de la couche routage seront expliquées dans la section 3.8, après nous présentons dans la deuxième partie l'attaque de type *black hole* et on termine par une conclusion.

3.2 Définition de Routage

Le routage c'est une méthode la plus important dans les réseaux, elle sert à l'acheminement des informations à partir d'un nœud source vers un nœud destination via un réseau de connexion donné.

L'objectif d'un protocole de routage est de garantir que les données ont atteint la bonne destination avec le moindre coût en terme de charge. Ce mécanisme assure la connectivité et la maintenance des liens dans le réseau en cas des ruptures.

Si on considère le nombre de saut comme étant un critère de sélection de route optimal, le chemin indiqué dans la figure suivante est le chemin optimal reliant le nœud source et le nœud destination. Une bonne stratégie de routage utilise ce chemin dans le transfert des données entres les deux nœuds.

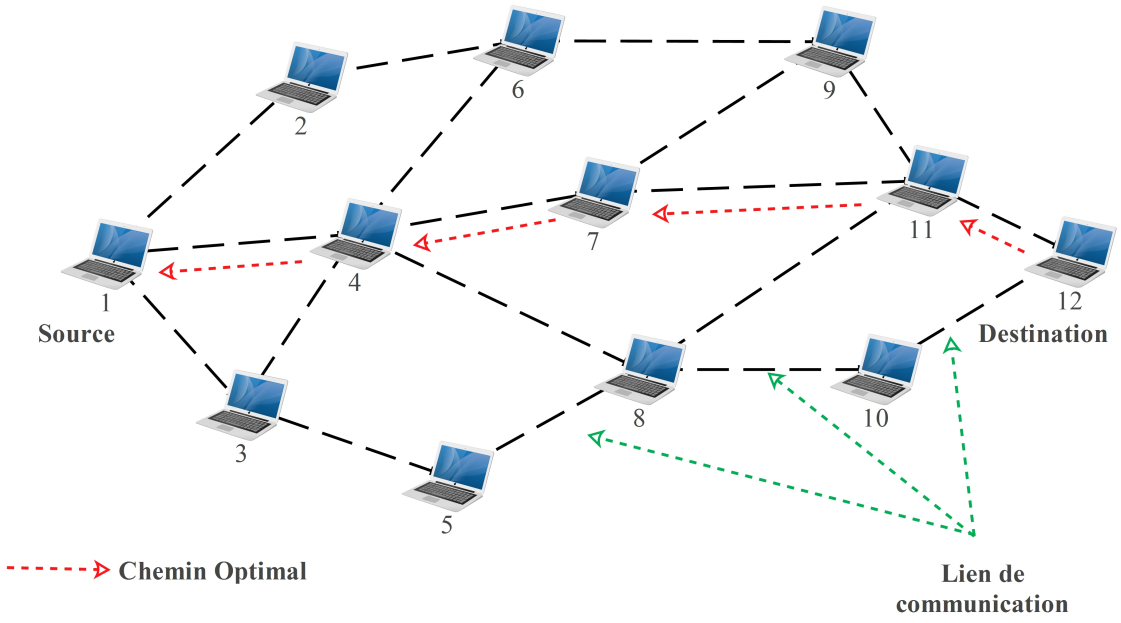

Figure 3.1 – Chemin optimal utilisé dans le routage entre la source et la destination.

3.3 Problématiques de routage dans les réseaux Ad-hoc

Les réseaux ad-hoc se caractérisent par l'absence d'infrastructure fixe ainsi que les nœuds sont mobiles, pour cela le routage est un challenge pour les chercheurs dans le but d'assurer la connectivité entre les nœuds. Chaque nœud est susceptible de collaborer pour participer au routage et pour retransmettre les paquets d'un nœud qui n'est pas en mesure dans la porté de sa destination, a ce point le nœud joue ainsi le rôle de station et de routeur au même temps.

La gestion de routage dans un environnement ad-hoc est différente par rapport aux approches utilisées dans le routage classique à cause de la taille du réseau qui peut être très grande, en outre ils sont susceptible à des changements fortes au niveau de topologie.

Le problème qui se pose dans ce contexte est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre d'unité existant dans un environnement caractérisé par la mobilité et la capacité de calcule. Pratiquement le nœud peut garder jusqu'à une quantité d'information de routage déterminé sur un nombre de nœud, donc il est impossible d'avoir des informations sur tous les réseaux.

les protocoles de routage doivent assurer que la diffusion du paquet de contrôle de routage soient moins coûteuse, car si ce trafique augmente peut dégradé les performances du réseau et plus dans le cas où il n'existe pas des informations sur la destination. Cette diffusion dans la découverte de route conduit à une surcharge dans le réseau qui est déjà caractérisé par une bande passante limité sans pris en compte en rajoutant les données transmises.

Généralement c'est rarement qu'on trouve le nœud destination dans la même zone de la source, dans ce cas le routage sera inutilisable donc le transfère de donné se fait directement. Pratiquement c'est totalement différent le routage devient nécessaire pour atteindre la destination, d'autre part il est très difficile pour adapter un tel protocole suite au changement inattendu du réseau.

3.4 Classification des protocoles de routage

L'idée principale est la création d'une architecture qui tient en compte des caractéristiques de réseaux ad-hoc en utilisant les différentes critères, donc la classification du protocole de routage (voire figure 3.2) se fait suite à plusieurs techniques et stratégies de découverte et le maintien du chemin ensuite leur façon de construction des tables de routage. Ainsi que les rôles attribués aux différents nœuds, enfin on peut

y classifié suivant [Akkaya and Younis, 2005] :

1. Architecture (plate ou hiérarchique ou géographique).
2. Algorithmes.
3. Catégorie (proactifs, réactifs ou hybrides).

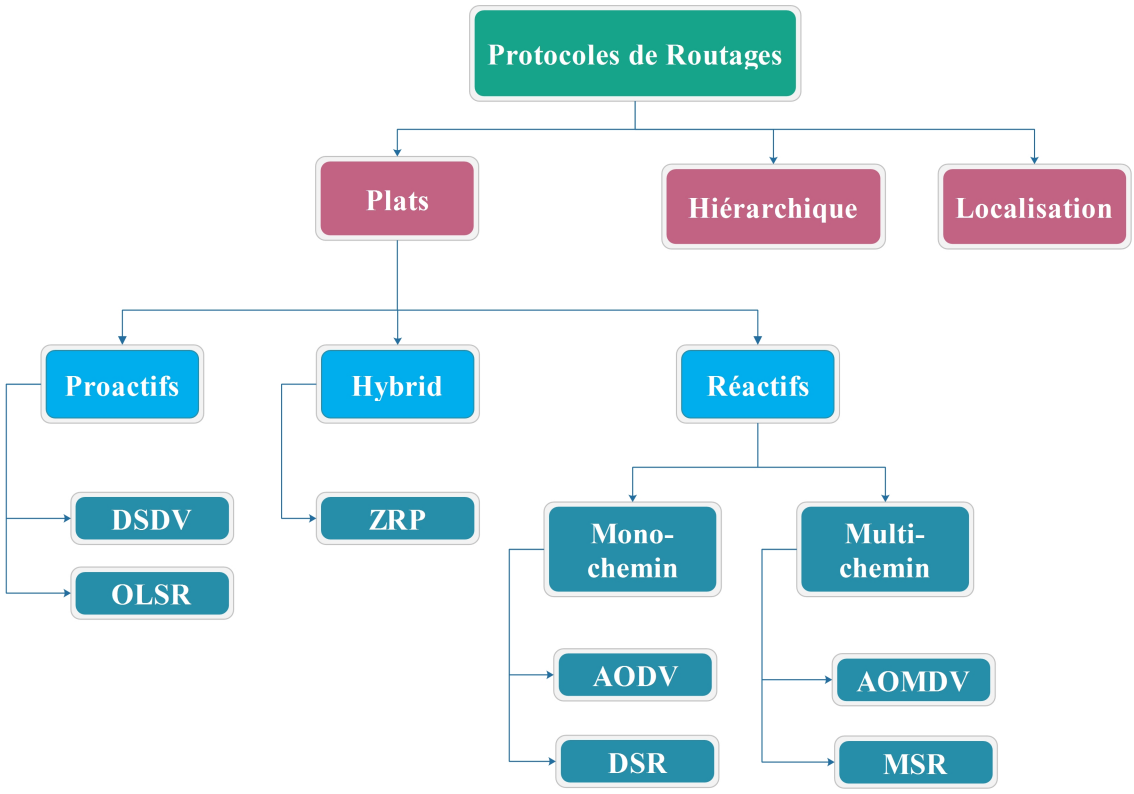

Figure 3.2 – Classification des protocoles de routage pour les réseaux ad-hoc.

3.4.1 Classification suivant le groupe MANET

C'est la classification la plus utilisée et qu'on détaillera dans la suite de ce chapitre. Suivant la manière de création et de maintenance des routes lors de l'acheminement des données, les protocoles de routage peuvent être classifié en : Proactif, Réactif et Hybride.

3.4.2 Les protocoles de routage proactifs(Avant la demande)

Dans cette catégorie du protocole les tables de routage sont construites au préalable, donc chaque nœud crée sa table avant que la demande ne soit faite. La topologie est connue à chaque instant, encore chaque nœud échange périodiquement des paquets de contrôles (voir figure 3.3). Les procédures de création et de maintenance

des routes, reste toujours active même s'il n'y a pas des paquets de données circulant dans le réseau.

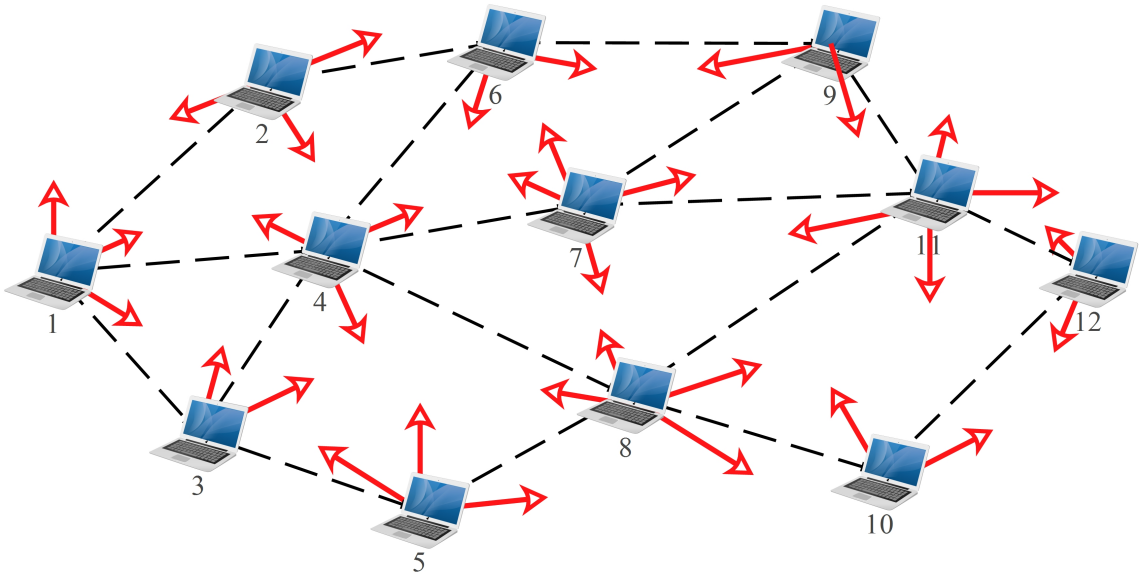

Figure 3.3 – Envoi périodique d'information topologique.

Deux principales méthodes sont utilisées dans cette classe de protocoles proactifs :

- la méthode État de Lien (*Link State*).
- la méthode du Vecteur de Distance (*Distance Vector*).

Ces méthodes sont utilisées aussi dans les réseaux filaires. Les algorithmes de routage basés sur ces deux méthodes, utilisent la technique des plus courts chemins, et permettent à un nœuds donné, de trouver le prochain saut pour atteindre la destination en utilisant le chemin le plus court existant dans le réseau. Généralement le calcul du plus court chemin entre deux nœuds, est basé sur le nombre des nœuds intermédiaires (on dit aussi le nombre de sauts) que comportent les différents chemins qui existent entre les deux nœuds.

Les nœuds enregistrent des routes vers tous les destinations du réseaux même si ces routes sont inutilisables, cela est assuré par une mise à jour périodique des informations de routage qui doit être diffusée par les différents nœuds de routage du réseau. Parmi les protocoles de routages proactifs les plus connus on citera le DSDV [Perkins and Bhagwat, 1994], OLSR [Jacquet et al., 2001].

Le routage par État de Lien

Le routage à état de lien permet à chaque nœud de garder une vision générale sur les liens du réseaux, cela se fait à travers des inondations périodiques des infor-

mations par chaque nœud sur ses liens directement connecté avec leurs voisins. en d'autre terme suite aux informations obtenus, le nœud connait toute la topologie du réseau ainsi les états des liens qui le constituent.

Même dans le cas où il y a un changement d'état des liens. Un nœud qui reçoit les informations concernant le nouvel état des liens, mis à jour sa vision de la topologie du réseau.

la recherche du chemin optimal qui est menée vers la destination se fait par l'un des algorithmes de plus court chemin prenant comme exemple celui de Dijkstra [Karp and Kung, 2000]. Le nœud applique cet algorithme de calcul des chemins optimaux sur toutes la topologie avec leurs liens récents afin de choisir le nœud suivant pour atteindre une destination donnée. Cela veut dire que pour qu'un nœud S puisse déterminer le nœud de routage suivant pour une destination, p doit recevoir les messages de la dernière mise à jour des liens, propagés par le réseau. Le protocole OSPF (Open Shoretest Path First) [Fortz and Thorup, 2000], est l'un des protocoles les plus populaires basé sur le principe *Etat de lien*.

Le routage par Vecteur de Distance Distance Vector

Dans l'approche de routage *Distance Vector*, chaque nœud calcul la distance entre les autres nœuds du réseau à travers les informations obtenues par les voisins. Chaque nœud informe ses voisins par sa distance avec ses nœuds proches. Suite aux informations diffusées par tous ses voisins, chaque nœud de routage fait un certain calcul pour trouver le chemin le plus court vers n'importe quelle destination.

Dans le cas où il y a un changement dans la distance minimal entre deux nœuds, cela fait appel à l'algorithme de calcul de chemin pour mettre à jour la distance vers les nœuds du réseau.

Cette technique est basée sur l'algorithme distribué de Bellman-Ford (DBF) [Awerbuch et al., 1994].

L'algorithme DBF est basé sur l'utilisation des messages de mise à jour. Un message de mise à jour contient un vecteur d'une ou plusieurs entrées dont chaque entrée contient, au minimum, la distance vers une destination donnée.

Le principe du DBF est appliqué par la majorité des protocoles de routage. Il utilise les vecteurs de distance minimale entre les nœuds comme étant une entrée. Le problème majeur de cet algorithme est la complexité temporelle, il prend beaucoup de temps pour mettre à jour les tables de routage des nœuds surtout dans le cas où il y a un grand nombre des nœuds dans le réseau.

Exemple Le protocole DSDV (Dynamic Destination-Sequenced Distance-Vector)

Le protocole DSDV [Perkins and Bhagwat, 1994] est principalement inspiré de l'algorithme distribué de Bellman Ford (DBF : Distributed Bellman-Ford). Toute fois, chaque nœud mobile se voit maintenir une table de routage contenant :

- Toutes les destinations possibles dans le réseau.
- Le nombre des nœuds (ou des sauts) nécessaire pour atteindre chacune de ces destinations.
- Le numéro de séquence (SN : Séquence Number) qui correspond au nœud destination.

Suite à la forte mobilité, la topologie change souvent, donc pour assurer la cohérence de table de routage, à chaque nœud est attribué un numéro de séquence NS qui permet de distinguer entre les nouvelles routes et les anciennes. Ce qui permet de résoudre le problème de la boucle de routage. Ainsi chaque nœud transmet à son voisin direct sa table de routage périodiquement ou en cas de changement inattendu de la table.

Donc, la mise à jour se fait selon deux facteurs : le temps et les événements qui peuvent surgir (déplacement des nœuds, apparition d'un nouveau voisin ...). Vu ces deux facteurs, on peut distinguer deux types de mise à jour :

- Mise à jour complète : Dans ce type le nœud transmet la totalité de sa table de routage vers ses voisins.
- Mise à jour partielle : dans cette mise à jour seulement une entrée concernée dans la table de routage qui doit être changée, cette entrée correspond au nœud qui est affecté par le nouveau événement (apparition d'un nouveau voisin, disparition d'un nœud ...). Notons que la mise à jour se fait à travers la transmission d'un paquet généralement contenant :
  - Le nouveau numéro de séquence, incrémenté, du nœud émetteur.
  - L'adresse de la destination.
  - Le nombre de sauts séparant le nœud de la destination.
  - Le numéro de séquence (des données reçues de la destination) tel qu'il a été reproduit par la destination.

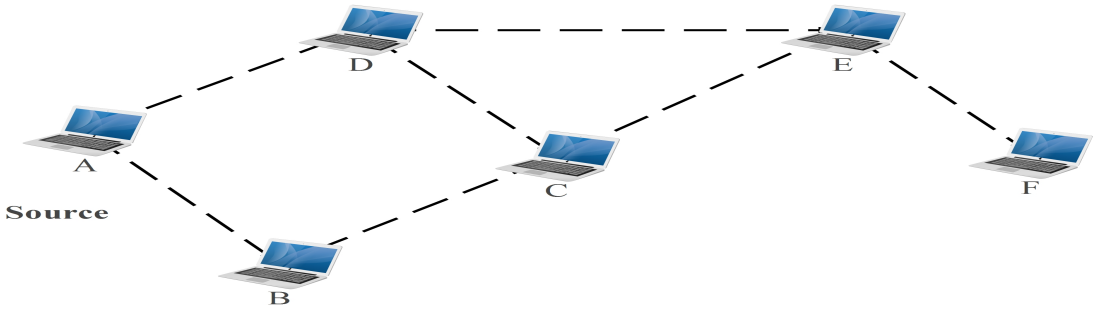

Figure 3.4 – Exemple de réseaux ad-hoc.

Si l'on considère que le DSDV est le protocole de routage utilisé dans la figure 3.4 la table de routage correspondante au nœud A ressemblera à la suivante :

Table 3.1 – Table de routage correspondante au nœud A

destination	Nombre de saut	Prochain nœud	numéro de séquence
A	0	A	NS1
B	1	B	NS2
C	2	B	NS3
D	1	D	NS4
E	2	D	NS5
F	3	D	NS6

Chaque nœud affecté par la mise à jour vérifie les donnée de routage reçu par les voisins, la route qui a le plus grand numéro de séquence sera prise en considération, Si deux routes ont le même numéro de séquence, alors la route qui possède le nombre minimum de sauts intermédiaires existants sur ce chemin est celle qui sera utilisée. Une valeur infinie est définie comme étant la métrique pour représenter les liens détruit, cette valeur est plus grande que la valeur maximale. Parmi les inconvénients du protocole DSDV, est qu'il est très lent, du fait qu'il doit attendre la mise à jour transmise par le destinataire pour modifier l'entrée adéquate dans la table de distance. Bien qu'il remédie au problème de boucle de routage *Routing Loop* et du *Counting to Infinity* (du DBF).

Avantages et inconvénients des protocoles proactifs

L'avantage d'un protocole de routage proactif est le gain de temps lorsqu'une route est demandée. En effet, les protocoles proactifs permettent de maintenir une table de routage à jour par l'échange périodique des messages. Ces tables étant à jour, car l'envoie de ces messages se fait rapidement. Cependant, on ne peut pas nier que l'émission régulière de ces paquets occupe une partie de la bande passante, qui risque d'augmenter en fonction du nombre des nœuds présents sur le réseau.

3.4.3 Les protocoles de routage réactifs à la demande

Les protocoles de routage réactif (protocoles de routage à la demande) représentent la majorité des protocoles les plus récents proposés dans le but d'assurer le service de routage, Les protocoles de routage appartenant à cette catégorie, créent et maintiennent les routes selon les besoins. Lorsque un nœud a besoin d'une route, il lance une procédure de découverte globale de routes, et cela dans le but d'obtenir une information spécifiée à une destination au préalable.

Dans ce cadre plusieurs politiques peuvent être adoptées, les plus importantes sont :

La technique d'apprentissage en arrière

Le mécanisme d'apprentissage en arrière ou le *backward learning* est basé sur le fait que lorsqu'un nœud source veut transmettre un message à une destination bien précise, Il est précédé tout d'abord à l'opération d'inondation de sa requête sur tout le réseau. Chaque nœud intermédiaire dit de transit (appartenant au chemin par lequel va passer le message), répond au nœud source lors de la réception de la requête par des informations sur le chemin recherché.

L'intermédiaire apprend le chemin qui est mené vers le nœud source, tout en sauvegardant la route dans la table transmise. Enfin, lorsque la requête arrive à la bonne destination, le nœud destinataire, suit le même chemin pour transmettre sa réponse sous forme d'un paquet de réponse.

Notons que le chemin établi entre les nœuds est un chemin bidirectionnel. Généralement aussi la source garde la trace du chemin tant qu'il restera en cours d'utilisation une fois que le chemin sera calculé.

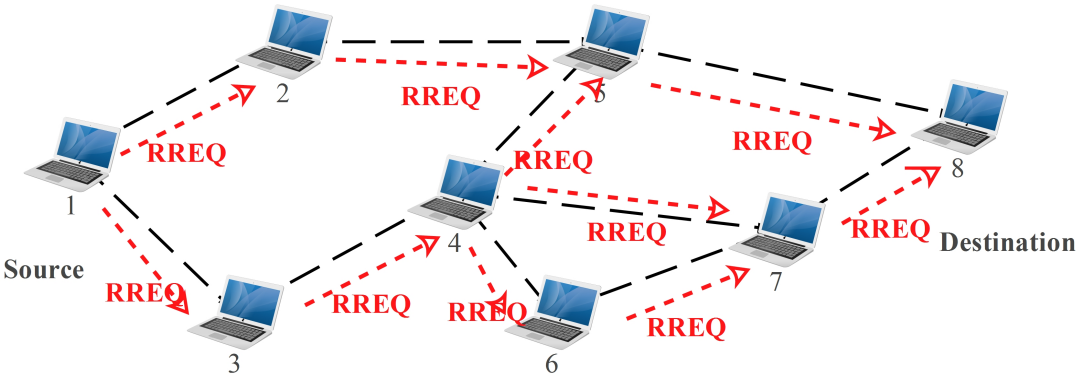

Figure 3.5 – Emission d'un REEQ

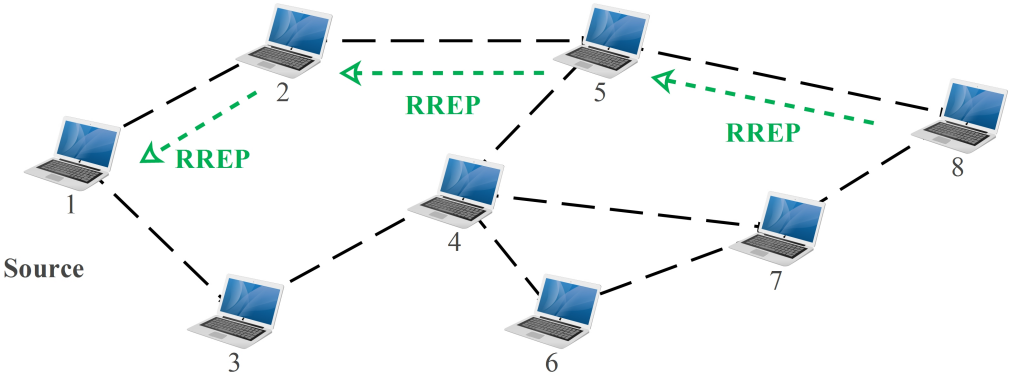

Figure 3.6 – Emission d'un RREP

La technique de routage source

Dans cette technique, le nœud source détermine toute la liste des nœuds par lesquels doit passer le message, ainsi le nœud émetteur inclut dans l'entête du paquet de cette liste.

En effet, afin de construire la route, le nœud source doit préciser les adresses exactes des nœuds par lesquels le message transitera jusqu'à atteindre le destinataire. Ainsi, le nœud source transmet le paquet au premier nœud spécifié dans la route.

Notons que chaque nœud par lequel le paquet transite, supprime son adresse de l'entête du paquet avant de le retransmettre. Une fois que le paquet arrive à sa destination, il sera délivré à la couche réseau du dernière hôte.

Plusieurs protocoles de routage réactif existent dont l'AODV, TORA, DSR.

Exemple Le protocole DSR (Dynamic Source Routing)

Le protocole de routage DSR, qui signifie *Dynamic Source Routing*, utilise la technique du routage source [Tuteja et al., 2010; Mittal and Kaur, 2009] . Le routage source consiste à ce que la source détermine un chemin et envoie dans chaque paquet de données tous les nœuds à traverser pour atteindre la destination. Chaque nœud intermédiaire retire son adresse du paquet avant de le retransmettre.

Cette technique nécessite la connaissance de la route à utiliser de la part de la source. Cette connaissance des routes est obtenue par une table de routage maintenue dans chaque nœud. Il faut donc dans un premier temps découvrir les routes, puis les conserver tant qu'elles existent.

Mécanisme de découverte ded routes (Route Discovery) Pour établir ces routes, chaque nœud peut initier une découverte dynamique de route. Pour cela le nœud qui lance une telle procédure va inonder le réseau d'une requête découverte

de route qui identifie la source. Si la requête parvient jusqu'à la destination, celle-ci renvoie le paquet à la source. Le paquet contient la liste des nœuds à traverser pour l'atteindre. En plus de l'adresse de la source le paquet contient la liste de tous les nœuds jusqu'à présent visité, ainsi chaque nœud qui reçoit le paquet peut créer à partir de celui-ci une table de routage qu'il pourra par la suite l'utiliser.

Chaque paquet de requête de route contient un identificateur unique permettant de détecter les duplications de ce paquet. Chaque nœud du réseau maintient ainsi une liste de couple $\langle adresse\ de\ l'initiateur,\ identificateur\ de\ requête \rangle$ des requêtes reçues, chaque entrée de la liste maintenue pour un temps de vie limité. Lors de la réception d'un paquet de requête de route par un nœud X du réseau, les opérations suivantes sont effectuées :

- si l'adresse de l'initiateur et l'identifiant de requête du paquet reçu existe déjà dans la liste des requêtes récemment reçues, le paquet est ignoré.
- Dans le cas contraire, si l'adresse de X existe dans le champ enregistrement de route du paquet de la requête, le paquet est ignoré.
- Sinon, si l'adresse de X est la même que l'adresse de la destination, alors l'enregistrement de route (contenu dans le paquet de la requête) contient le chemin à travers lequel le paquet de la requête est passé avant d'atteindre le nœud X.

Une copie de ce chemin est envoyée dans un paquet de réponse de route vers la source. Sinon, l'adresse de nœud X est ajoutée dans l'enregistrement de route du paquet reçu, et le paquet est rediffusé.

Le nœud A veut trouver une route vers le nœud O, On voit ici le principe de découverte de route par DSR Pour retourner le paquet, la destination utilise un chemin qu'elle connait déjà, si elle ne possède pas de chemin pour joindre la source elle peut utiliser le chemin qui se trouve dans le paquet qu'elle a reçu, si l'environnement le permet. En effet dans certains réseaux les nœuds ne sont pas forcément bidirectionnels.

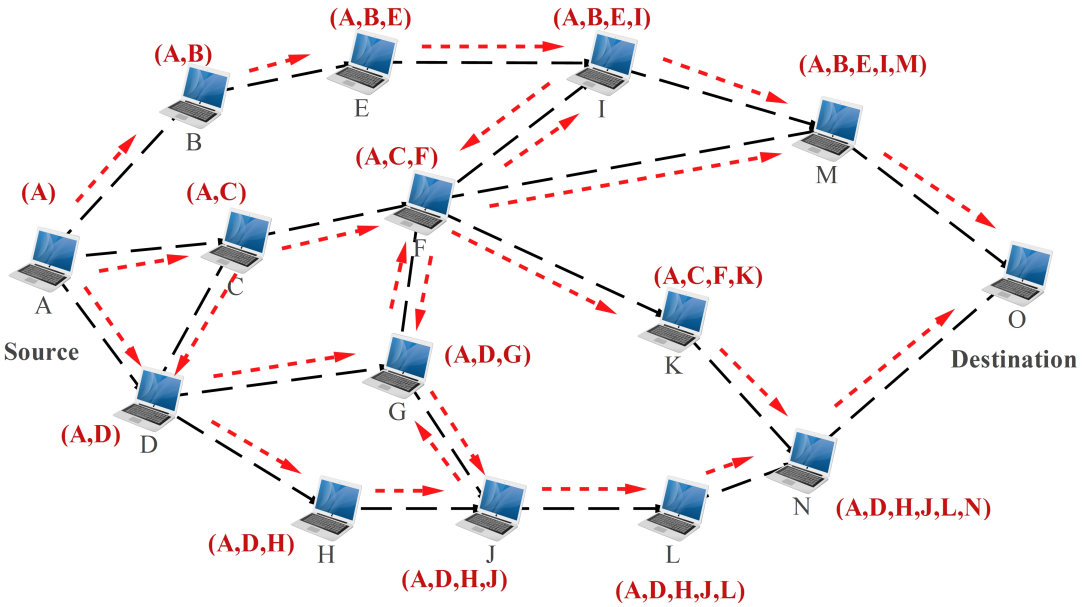

Figure 3.7 – mécanisme de découverte de routes dans DSR (Route Discovery)

Le mécanisme de la maintenance de route Le protocole DSR maintient les routes et l'utilisent jusqu'à ce qu'il y a une coupure de lien détectée.

la défaillance est déterminée par les retransmissions de la couche liaison de données, quand un nœud du chemin détecte une coupure, ce dernier envoie un message d'erreur à la source pour dire que le chemin n'est plus valide après lui. A ce moment la source peut adapter sa table de routage et doit relancer une nouvelle requête de découverte de route.

L'avantage du protocole DSR L'avantage principal de DSR réside dans sa faiblesse d'échange des paquets des contrôles quand peu de sources communiquent avec les destinations rarement accédées. Mais peut avoir le problème de grande échelle (*scalability*).

De plus, les nœuds intermédiaires ne maintiennent pas la table de routage pour les paquets qu'ils reçoivent sachant que ces derniers possèdent déjà toutes les décisions de routages, Même aussi DSR évite les boucles de routage parce que le chemin est mentionné dans les paquets. Cependant le chemin utilisé n'est pas forcément optimum, et la découverte d'une route peut prendre du temps.

Quand le réseau devient plus grand, les paquets deviennent plus grands puisqu'ils doivent contenir des adresses du nœud dans le chemin.

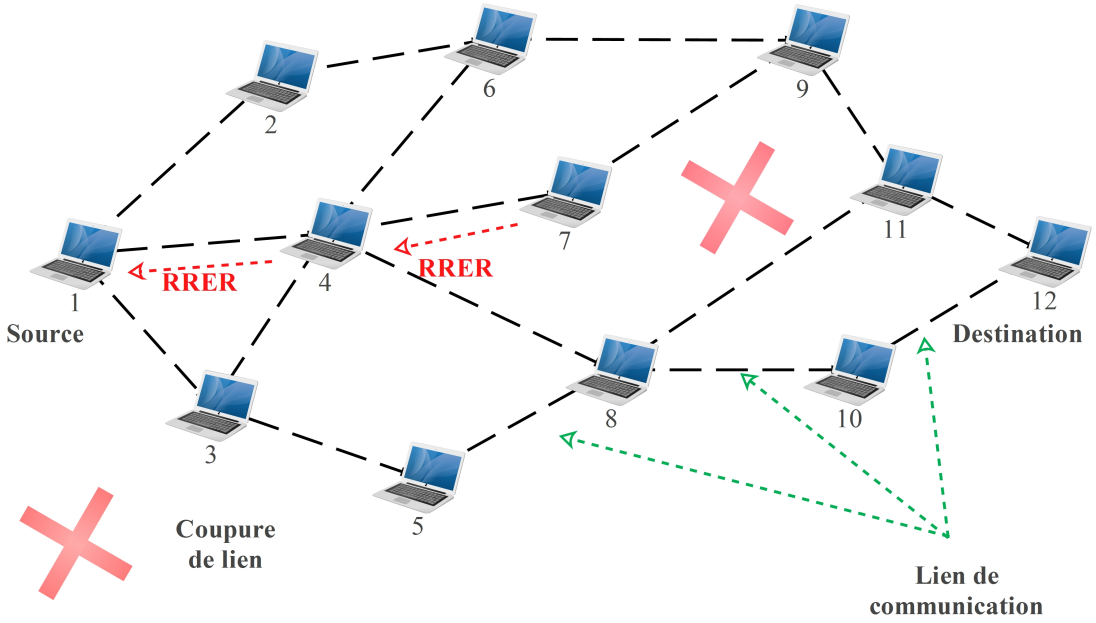

Figure 3.8 – Le mécanisme de maintenance de route dans DSR

Avantages et inconvénients de protocole réactif

Le routage à la demande induit une lenteur à cause de la recherche des chemins, ce qui peut dégrader les performances des applications interactives (exemple les applications des bases de données distribuées). En outre, il est impossible de connaitre au préalable la qualité du chemin (en termes de bande passante, délais, etc.). Une telle connaissance est importante dans les applications multimédias.

3.4.4 Les protocoles de routages Hybrides

Les protocoles hybrides combinent les deux idées : celle des protocoles proactifs et celle des protocoles réactifs. Ils utilisent un protocole proactif pour avoir des informations sur les voisins les plus proches (au maximum les voisins à deux sauts).

Entre les zones prédéfinies, le protocole hybride utilise les techniques des protocoles réactifs pour chercher des chemins.

Ce type de protocoles s'adapte bien aux grands réseaux, cependant, il cumule aussi les inconvénients des protocoles réactifs et proactifs en même temps (messages de contrôle périodique, le coût de découverte d'une nouvelle route).

Plusieurs protocoles hybrides existent dont le CBRP *(Cluster Based Routing Protocol)* [Ibriq and Mahgoub, 2004] et le ZRP (Zone Routing Protocol) [Haas et al., 2002].

Le protocole ZRP (Zone Routing Protocol)

Le protocole de routage ZRP [Haas et al., 2002] utilise les deux approches (Proactif et Réactif),

Il crée des zones de routage ou les nœuds voisins utilisent les protocole proactive uniquement (les changements de la topologie doivent avoir un impact local dans le coût d'échange du message de contrôle),

La partie hors la zone de routage (les zones voisines) offre une recherche rapide et efficace dans le réseau, Contrairement à une recherche sur tout le réseau, dans ce protocole, la détection des boucles de routage est possible grâce à la connaissance de la topologie du réseau.

chaque nœud fait partie d'une zone de routage, elle inclut les nœuds qui sont à une distance minimale (en termes de nombre de sauts), du nœud en question, inférieure ou égale au rayon d de la zone.

La figure 3.9 illustre la zone associée au nœud A avec un rayon de deux sauts.

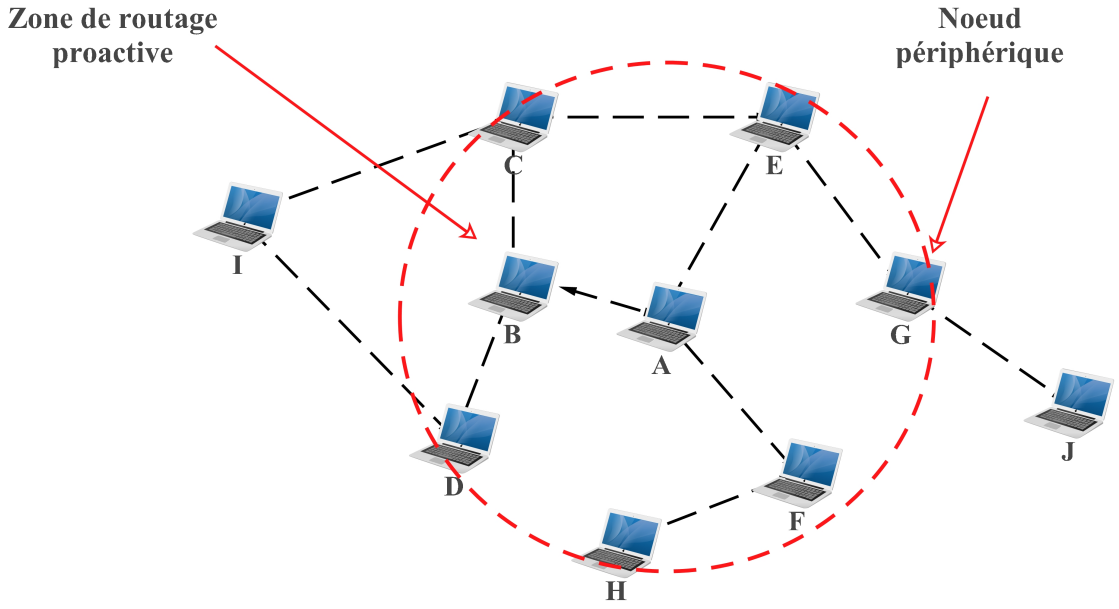

Figure 3.9 – la zone de routage A avec d=2.

Les nœuds dans ZRP sont divisés en trois types :

1. Les nœuds internes : ce qui appartiennent aux zones de routage sachant que la distance qui les séparent par le nœud centre A de la zone est strictement inférieure au rayon d. exemple B, E,et F
2. Les nœuds périphériques : ce qui situent dans une distance égale de rayon C, D, H, et G

3. Les nœuds externes : ce sont hors la zone de routage I et J,

En résumé, ZRP définit donc trois types de protocole [Beijar, 2002] : l'un fonctionne localement et le deuxième fonctionne entre zones. Ces deux protocoles sont :

- IARP (IntrAzone Routing Protocol) fournir des routes optimales vers les nœuds qui se trouvent à l'intérieur de la zone à une distance fixe, et tout changement est considéré uniquement dans la zone.
- IERP (IntErzone Routing Protocol) quant à lui s'occupe de rechercher les routes à la demande pour des destinations à l'extérieure d'une zone.
- BRP (Bordercast routing protocol) : permet de construire les nœuds de périphérie et la façon de les atteindre à travers les données de la topologie fournies par le protocole IARP. Il est utilisé pour acheminer les requêtes de découverte de route de l'IERP dans le réseau.

De plus, ZRP suppose que chaque nœud connait les voisins de sa zone à travers IERP, donc si la destination se trouve dans la même zone le chemin est connu. Avant que la découverte de chemin est lancé ZRP vérifie si le nœud destinataire ne se trouve pas dans la zone du nœud source.

Autrement, une demande de découverte de route *RREQ* est envoyée vers tous les nœuds périphériques, ces derniers vérifient, à leur tour, si la destination spécifiée par la source existe dans leurs zones. Si la destination est située dans la zone, le nœud périphérique envoi un paquet *RREP* contenant le chemin menant à la destination, sinon, les nœuds périphériques diffusent la *RREQ* à leurs propres nœuds périphériques, qui à leurs tours, effectuent le même traitement.

Dans la Figure 3.9 , le nœud A veut envoyer un paquet au nœud J, puisque ce dernier n'est pas dans la zone de routage de A, une requête *RREQ* est envoyée par A aux nœuds périphériques qui sont C, D, G, et H. Ces derniers vérifient l'existence du nœud J dans sa zone de routage et par l'envoie d'un message *RREP* contenant le chemin établi du nœud G.

3.5 Spécification du protocole de routage AODV

3.5.1 Principe de fonctionnement

AODV est un algorithme de routage conçu par [Perkins and Royer, 1999; Chakeres and Belding-Royer, 2004]. Il est adapté aux réseaux de topologie fortement dynamique et il est basé sur le routage à vecteur de distance.

Le protocole AODV crée les routes à la demande (s'il a besoin d'un chemin), il permet de réduire le nombre de messages de contrôle diffusé dans le réseau, c'est-

à-dire qu'il la construction des routes se fait lorsqu'elle est demandée par le nœud source. De plus les routes sont maintenues durant le temps de leur utilisation.

le routage se fait nœud à nœud (multi-sauts) tous les nœuds de réseau doivent participer dans la recherche de chemin. AODV utilise le principe des numéros de séquence qui permettre aux nœuds d'utiliser les routes les plus fraiches (voir détail dans la section 3.5.2).

L'établissement et la maintenance des routes sont assurés par l'échange de différents types de paquets contrôle [Perkins et al., 2003] :

1. Route Request *RREQ* "J'ai besoin d'une route "le paquet diffusé à tous les nœuds voisins par le nœud source voulait envoyer des paquets de données vers une destination.

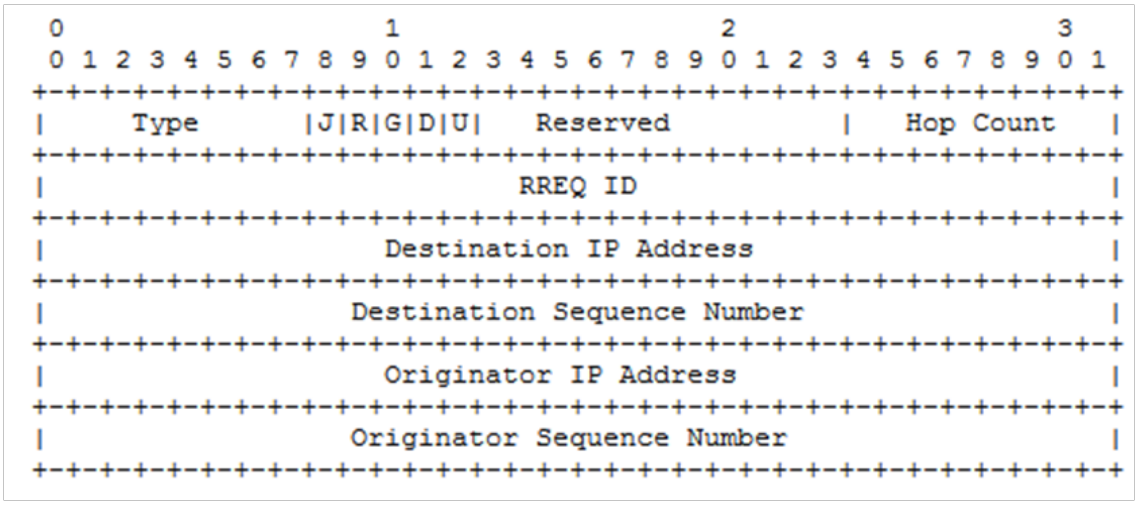

Figure 3.10 – Format général d'un RREQ

2. Route Reply *RREP* "répondre la route", une fois la destination reçoit le RREQ, elle répond par un *RREP* comme accusé de réception, qui contient le chemin inverse de *RREQ* vers la source.
3. Hello message "Vous êtes là ?", c'est un message diffusé périodiquement vers le nœud immédiatement voisin pour voir s'il est encore dans la même zone, s'il n' y a pas de message Hello qui arrive d'un nœud particulier, le voisin suppose que ce nœud est déplacé et marque ce lien non disponible.
4. Route Error "Annuler la route", c'est un message envoyé par un nœud lorsqu'il détecte que la liaison avec son voisin est cassée (route invalide).

Plusieurs procédures d'établissement et de maintenance de route ont été introduites dans AODV sont :

– Découverte de route (Path Discovery).

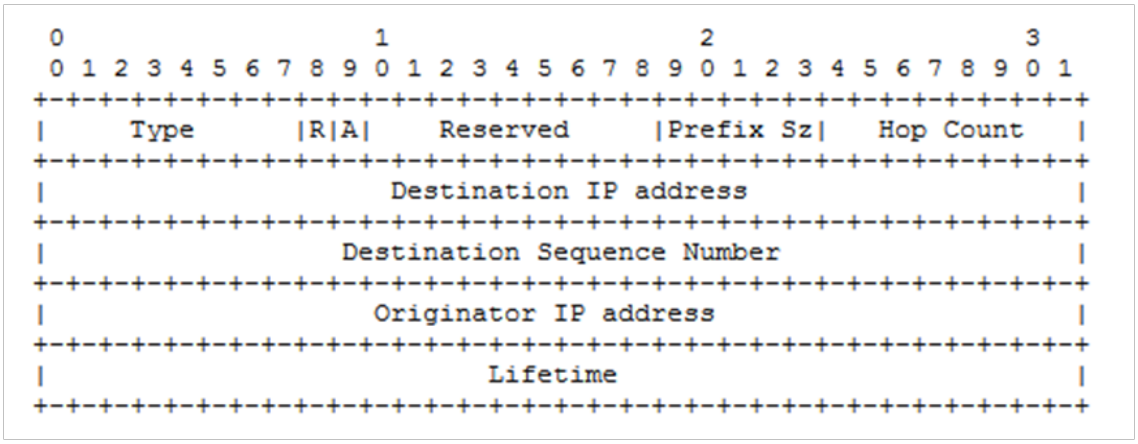

Figure 3.11 – Format général d'un RREP

- Création de route inverse (Reverse Path Setup).
- Acheminement de *RREP* (Forward Path Setup).
- Maintenance de route (Path Maintenance).
- ôconnectivité locale (Local Connectivity Management).

Ainsi, le mécanisme de routage du protocole AODV repose sur le principe suivant : "échanger des paquets de routage bien précis pour trouver une route et puis la maintenir durant son utilisation ". Chaque nœud à la réception d'un paquet de routage doit apporter les modifications nécessaires à sa table de routage et puis décider ce qu'il doit en faire. L'établissement d'une route suit un cycle Requête/Réponse.

3.5.2 Numéros de séquence

Numéros de séquence permet aux nœuds de comparer le degré de la fraicheur de leurs informations de routage sur les autres nœuds. Toutefois, lorsqu'un nœud envoie un des messages de contrôles(RREQ, RREP, *RERR* ...), il augmente sa propre valeur de numéro de séquence.

Une valeur de numéro de séquence supérieur est une information plus précise elle permet d'identifier le nœud qui envoie cette information. Le nœud propriétaire de ses informations est considéré comme le saut suivant pour l'établissement de la route vers la destination

Le numéro de séquence est une valeur sur 32 bits non signée (valeur max=4294967295). Si le numéro de séquence du nœud atteint la valeur la plus élevé, alors il sera remis à zéro, tandis qu'à chaque nouvelle tentative d'émission cette valeur incrémentée par 2.

3.5.3 La gestion de la table de routage

AODV maintient les chemins d'une façon distribuée, chaque nœud intermédiaire fait partie du chemin recherché, stocke une entrée dans sa propre table de routage. Chaque entrée de la table de routage contient les informations suivantes :

- Adresse du nœud destination : c'est l'adresse IP du nœud destinataire à atteindre.
- Adresse du nœud suivant : l'adresse IP du nœud auquel on va envoyer un paquet à router pour joindre une destination.
- Nombre de sauts séparant le nœud source du nœud destination.
- Numéro de séquence associé à la destination.
- Durée de vie pour laquelle la route reste à la disposition du nœud source.
- Liste des voisins qui utilisent cette route : adresses IP d'éventuels nœuds précurseurs qu'utilise le nœud courant comme un prochain saut pour atteindre la destination.
- Un tampon de requête afin qu'une seule réponse soit envoyée par requête.

A chaque utilisation d'une entrée, son temps d'expiration est mis à jour (temps courant + temps actif route).

3.5.4 Mécanisme de découverte de route

Quand un nœud aurait besoin de connaitre un chemin vers une certaine destination il consulte sa table de routage, dans le cas où aucune route n'est disponible, la source diffuse une requête de route *RREQ* (Figure 3.12). La diffusion d'un paquet *RREQ* faite dans plusieurs cas, si la destination n'est pas connue au préalable, ainsi que si le chemin existant qui mène vers la destination a expiré sa validité ou s'il est devenu défaillant (i.e. la valeur de métrique qui lui est associée est infinie).

Dans la création d'un nouveau paquet *RREQ* la valeur de numéro de séquence destination est récupérée à partir de la table de routage, c'est la valeur de la dernière communication associée au nœud destination. Si cette valeur n'est pas connue, le champ numéro de séquence destination est initialisé par la valeur par défaut (nul). D'une autre part, Le numéro de séquence source du paquet *RREQ* contient la valeur du numéro de séquence du nœud source.

La source attend une période de temps déterminée ($RREP\_WAIT\_TIME$), après la diffusion du *RREQ* s'il y a pas de réponse *RREP*, la source peut rediffuser une autre fois une nouvelle requête *RREQ*. Le paquet *RREQ* aurait rediffusé en augmentant le nombre maximum de sauts, Aussi le champ Broadcast ID du paquet *RREQ* est incrémenté pour identifier chaque requête de route particulière associée

à une adresse source. Cette procédure est répétée jusqu'à une valeur de retransmission maximale de fois *RREQ_RETRIES* avant de déclarer que cette destination est injoignable, à ce stade un message d'erreur est délivré à la couche d'application. Afin de limiter le coût dans le réseau, AODV propose d'étendre la recherche progressivement. Initialement, la requête est diffusée à un nombre de sauts limités.

Quand un nœud intermédiaire retransmit le paquet *RREQ* à un voisin, il sauvegarde aussi l'identificateur du nœud à partir duquel il a reçu la première copie de la requête. Cette identité sert aussi à la construction du lien inverse (Figure 3.13), ces chemins sont utilisés pour acheminer le paquet réponse *RREP* d'une manière unicast (cela veut dire qu'AODV supporte seulement les liens symétriques), En outre l'identifiant du paquet *RREQ* assure la non utilisation du même paquet. Puisque le paquet *RREP* va être envoyé à la source, les nœuds appartenant au chemin de retour vont mettre à jour leurs tables de routage suivant le chemin contenu dans le paquet de réponse (temps d'expiration, numéro de séquence et saut suivant).

La destination renvoie un message *RREP*, ce message est acheminé vers la source. Chaque nœud traversé incrémentera le nombre de sauts cela permet au nœud source de déterminer le nombre de saut qui le sépare avec la destination, De plus il ajoutera une entrée à sa table de routage pour la destination (le nœud émetteur de RREP). Une réponse convenable peut aussi être transmis par un nœud intermédiaire situé entre la source et la destination (pas forcement la destination) s'il a une route valide dans sa table de routage et sa valeur de numéro de séquence est supérieure à la valeur de numéro de séquence du RREQ, de plus le nombre de saut doit être inférieur ou égal.

Après l'émission du paquet *RREP* le paquet *RREQ* est ignoré, seulement si le paquet *RREQ* contient un drapeau G , Dans ce cas le nœud intermédiaire envoyé alors en plus un *RREP* vers la destination "Gratuitous RREP". Les nœuds entre le nœud intermédiaire et la destination ajouteront donc à leur table une entrée vers la source du *RREQ*. Cette information permettra à la destination d'envoyer des paquets de donnée 'à la source sans devoir procéder à la recherche d'une route. C'est une technique pratique quand on a une communications TCP pour l'envoi du paquet d'acquittement *ACK*.

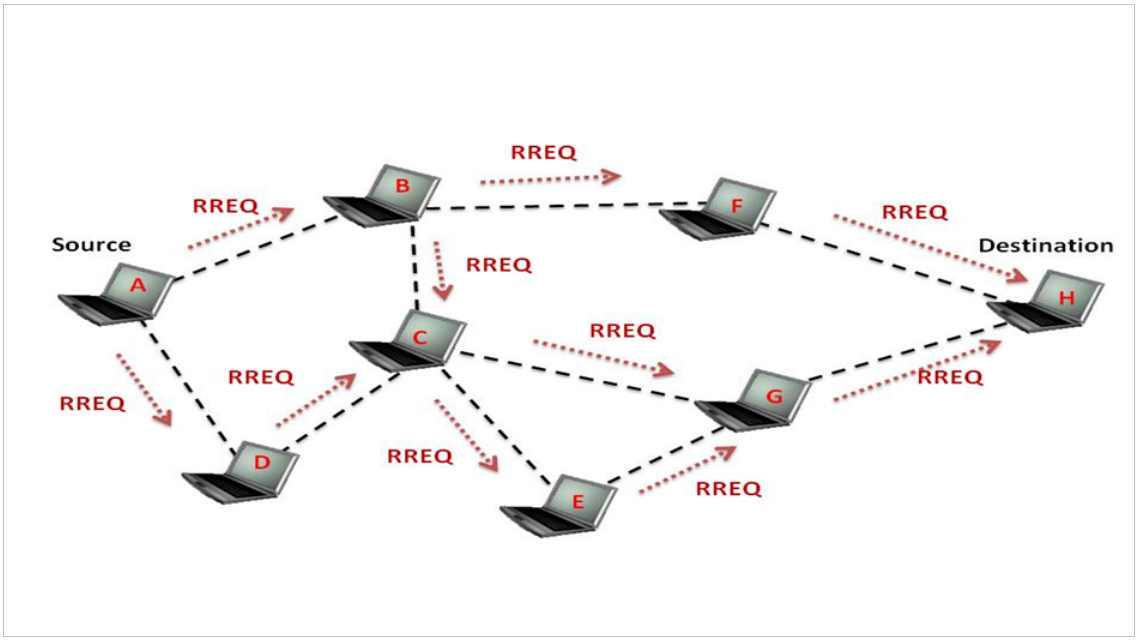

Figure 3.12 – Demande une route (RREQ)

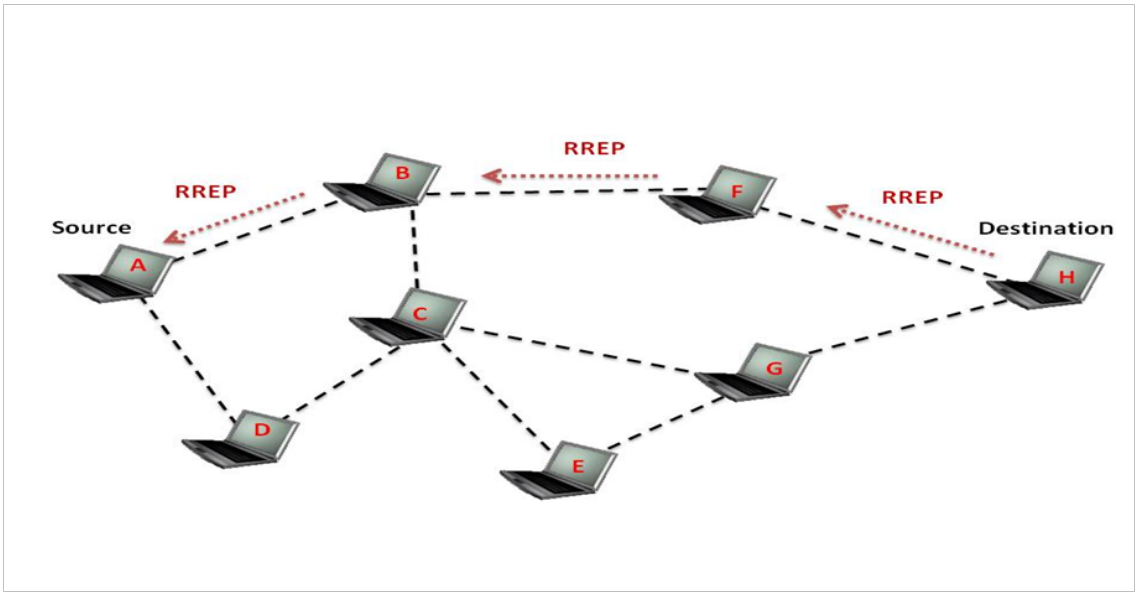

Figure 3.13 – Réponse de route (RREP)

3.5.5 Maintenance des routes

La maintenance des routes [Ibriq and Mahgoub, 2004] est une technique qui assure la disponibilité des liens, quand un chemin est établi entre un nœud source et destination il sera maintenu durant la communication. Cette technique est déclenchée automatiquement, si un nœud détecte la défaillance d'un lien entre le voisin suivant de la route donc il doit essayer de le réparer.

Cette procédure est composée de deux parties globales :

A) Gestion de la connectivité Chaque nœud dans AODV maintient une liste de ses voisins. Cette liste est construite à travers une diffusion d'un paquet de contrôle appelé *HELLO*. La diffusion est effectuée périodiquement dans un intervalle de temps fixe *HELLO_INTERVAL*. Ces paquet ne sont destinés qu'aux nœuds voisins (à un saut), car la valeur de *TTL* est égale à un pour éviter qu'il ne soit propagé plus dans le réseau. Chaque nœud appartient au chemin de routage actif doit surveiller la connectivité du lien avec leur nœud successeur de ce chemin, Quand il ne reçoit pas le paquet *HELLO* durant la période *HELLO_INTERVAL* en provenance du nœud voisin, le lien est considéré défaillant.

B) Les défaillances des liens : Elles sont généralement dues à la mobilité des nœuds ou à cause d'épuisement d'énergie. Si le nœud source qui se déplace hors sa zone de couverture, donc il se provoque une coupure sur la liaison avec son successeur, alors il relancera la procédure de découverte de route s'il a encore besoin d'un chemin. D'autre part, si le nœud qui se déplace est un nœud intermédiaire ou destination, alors le nœud source doit être informé par le message *RERR* qui doit être généré par le nœud qui détecte cette coupure précisément le nœud le plus proche pour les deux (voir figure 3.14). Le nœud détecteur peut lancer une réparation local, il est responsable pour trouver un autre chemin, sinon il envoi un paquet *RERR* vers la source. Tous les nœuds de la route défaillante seront informés sur la coupure du chemin, En recevant un *RERR*, un nœud marque la route vers cette destination (dont l'adresse figure dans le *RERR*) comme étant invalide, en mettant la valeur du champ de la distance correspondant à l'infini (Distance = infinie), et à son tour renvoie le *RERR* vers ses précurseurs sur cette route. Lorsque le nœud source reçoit le *RERR*, il entame alors un nouveau processus de découverte d'une nouvelle route s'il en a encore besoin.

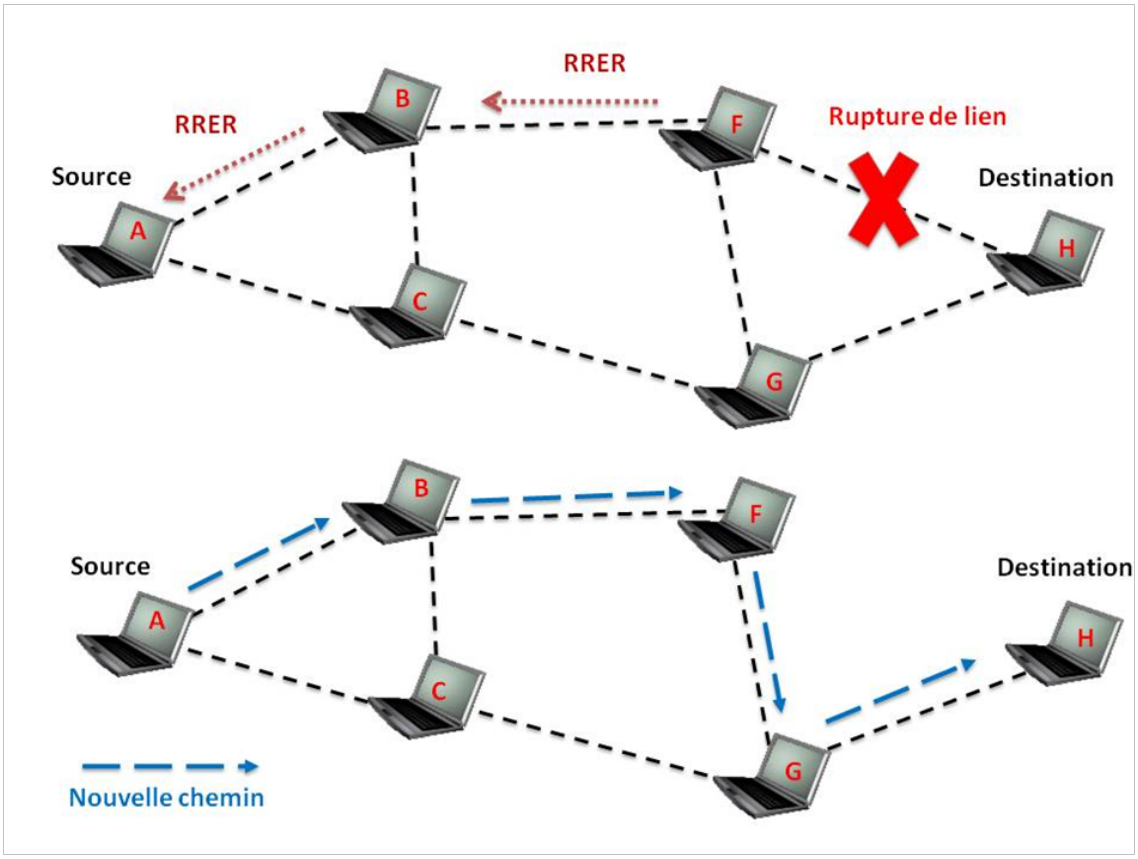

Figure 3.14 – Génération de *RERR* à cause de la défaillance du lien F-H.

3.5.6 Les Avantages et Inconvénients

AODV lance la découverte des route à la demande en inondant le réseau avec un paquet de requête donc il utilise moins de paquets de contrôles.

Le majeur avantage *d'AODV* se repose sur l'utilisation de numéro de séquence dans les paquets, Cette technique assure la non apparition des boucles infinies. De plus, il sert à identifier entre les routes fraiches qui sont essentielles au processus de mis à jour de la table de routage. L'identifiant du nœud source est redéfini dans chaque paquet, Ceci permet de ne pas perdre la trace du nœud source quand le paquet retransmit par les différents retransmetteurs.

Un inconvénient d'AODV est qu'il n'existe pas une format commun entre les paquets. Chaque paquet a son propre format : *RREQ, RREP, RERR*. D'une autre part, l'existence d'un délai avant que la transmission des données a été démarrée.

3.6 Sécurité des réseaux ad-hoc

Aujourd'hui, les réseaux filaires peuvent assurer un niveau de sécurité très élevé. Mais dans les réseaux sans fil Ad-hoc, les défauts de sécurités apparaissent souvent même si des précautions ont été prises. Ceci peut affecter les services qui tournent dans ce type de réseau, notamment les protocoles de routage du MANETs. Les vulnérabilités des réseaux Ad-hoc sont nombreuses et peuvent être classifiées suivant plusieurs critères et suite à certain modèle d'attaquant [Castelluccia et al., 2007; Sanzgiri et al., 2002; Hu et al., 2005].

Ces attaques sont généralement simples à exploiter relativement à une couche spécifique. La majorité des nœuds ont l'opportunité d'accéder à la zone de couverture des autres nœuds et donc lancer des attaques pour perturber le fonctionnement du réseau.

De plus, un attaquant qui connait bien les mécanismes et les protocoles de la couche réseau peut supprimer ou modifier les données de façon générale, notamment la coopération entre les attaquants qui conduit à un impacte très grave.

3.6.1 Les objectifs de la sécurité

Il existe plusieurs mécanismes pour protéger l'information et les ressources contre les attaques et les mauvais comportements dans un réseau mobile Ad-hoc [Gayraud et al., 2003], puisque ce type de réseaux a les mêmes services et fonctionnalitées que ceux des autres réseaux. Par suite, la sécurité dans les réseaux ad hoc vise les objectifs suivants [Lou and Fang, 2004] :

Authentification :

L'identité des nœuds doit être vérifiable ce qui interdit aux nœuds malveillants non authentifiés d'injecter des messages falsifiés (s'assurer qu'il n y a pas de nœud intrus qui est masqué usurpant l'identité d'un autre).

Confidentialité :

Le principe de la confidentialité assure que seuls les nœuds authentifiés communicants sont en mesure de comprendre les données secrètes échangées. Des méthodes de contrôles d'accès stricts doivent être mise en place pour assurer la confidentialité des données échangées au sein des nœuds réseaux .

Integrité :

L'intégrité des données garantit que les données échangées n'ont pas été altérées ou modifiées ou détruites durant la communication d'une façon volontaire ou accidentelle.

Non répudiation :

La non-répudiation assure qu'un message envoyé ne sera pas rejeté par son expéditeur, cette propriété est réalisée en appliquant une méthode basée sur la signature électronique.

La disponibilité :

Ce principe permet de s'assurer que les services réseaux désirés sont toujours disponibles. Le système qui assure la disponibilité dans un réseau Ad-hoc cherche à combattre les dénis de services et les nœuds malveillants tels que les nœuds égoïstes.

3.6.2 Classification des attaques

Après avoir présenté les notions de base relative au réseau sans-fil Ad-hoc, nous allons présenter maintenant les différentes catégories des menaces, des attaques et des vulnérabilités qu'on peut retrouver dans ce type de réseau [Rai et al., 2010]. L'utilisation des liens sans fil facilite les attaques contre les réseaux Ad-hoc que ce soit à travers une simple écoute ou d'attaque plus nuisible comme le dysfonctionnement intentionnel d'un service. La majorité des réseaux autonomes et auto-organisé est en particulièrement les réseaux sans fil Ad-hoc, se basent sur des algorithmes coopératifs nécessitant l'implication de tous les nœuds du réseau.

Les attaques sont classées en plusieurs critères suivants le modèle d'attaquant [Gagandeep and Kumar, 2012] :

Attaque passive ou active

Dans les réseaux Ad-hoc, selon le niveau d'intrusion et les actions menées par un attaquant, on distingue généralement deux catégories d'attaques : les attaques passives et les attaques actives.

Attaque passive : Va se limiter sur l'écoute non autorisé des flux et la surveillance des canaux de communication. Une écoute se produit lorsqu'un attaquant capture un nœud et étudie le trafic qui le traverse sans altérer le fonctionnement. L'intrus

essaye uniquement par ces écoutes d'obtenir et de stocker des informations qu'il n'est pas sensé connaitre ou garder dans le but de découvrir des informations de base tel que le protocole routage et les systèmes de sécurité pour obtenir des données privées d'un nœud ou d'un ensemble des nœuds sans affecter le déroulement de routage. Les données analysées aident l'intrus à agir plus tard. Un adversaire passif ne fait que menacer la confidentialité des données.

Attaque Active : Une attaque est active lorsqu'un nœud non autorisé altère des informations en transit par des actions de modification, suppression, ou fabrication, ce qui conduit à des perturbations du fonctionnement du réseau. D'une façon générale il modifie son comportement de façon arbitraire par rapport au comportement normal dans les différents protocoles dans lesquels il sera engagé. Il pourra aussi non seulement obtenir plus d'informations qu'un attaquant passif, mais aussi il pourra modifier significativement le fonctionnement d'un protocole ou d'empêcher son bon déroulement d'exécution. Les attaques actives sont classifiées en quatre groupes :

- Par suppression (drooping attacks) : le nœud malveillant supprime tous les paquets qui ne sont pas destinés à lui, et la majorité des protocoles de routage ne possède aucun mécanisme pour détecter si les paquets de données transmis ont atteint la destination ou non.
- Par modification (Modification Attacks) : le nœud malveillant capture les informations importante et essaye de modifier l'un des paramètres du protocole par exemple les informations de routage, ensuite, il annonce une information modifiée tel qu'il possède le plus court chemin à une destination .
- Par fabrication (Fabrication Attacks) : l'attaquant envoie des faux messages aux nœuds voisins sans recevoir des messages relatifs, par exemple l'attaquant envoie un paquets de réponse de route sans qu' il a reçu la requête de demande de route .
- Timming attacks : l'attaquant garde le paquet qu'il a reçu et qui n'y est pas destination et fait un retard pour le retransmettre après un certain temps ce qui provoque une perturbation dans l'échange des informations.

Attaque externes ou internes

Selon le domaine d'appartenance d'un nœud, les attaques actives peuvent elles mêmes être classées en deux catégories, à savoir les attaques externes et internes.

Un attaquant interne est celui qui arrive à contrôler le réseau, c'est à dire, un nœud interne ayant le statut d'un membre du réseau et qui dispose un ensemble de connaissance associe à ce statut (clé secret, table de routage ...). Les attaques

internes sont menées par des nœuds compromis qui sont autorisés à participer au fonctionnement du réseau. Les attaques internes sont généralement plus dangereuses et difficiles à détecter que les attaques externes.

Tandis que les attaques externe sont réalisée par des nœuds qui n'appartiennent pas au domaine de réseau. Un attaquant externe ne dispose pas à priori des connaissances mais il est capable d'effectuer des opérations cryptographiques pour empêcher le réseau de communication de produire une charge supplémentaire.

Attaque individuelle ou attaque distribuée

En effet, les attaques peuvent être de type individuelles ou par collusion avec d'autres participants qui sont appelées également distribuées. Les attaques individuelles sont menées par un seul nœud attaquant. Puisque les capacités de communication et de calcul de l'attaquant sont en général similaires à celles des autres nœuds du réseau, ces attaques demeurent relativement simples, et sont d'autant plus limitées par rapport aux mécanismes de sécurité qui sont mis en œuvre. En revanche, rien n'empêche à des nœuds attaquants de mutualiser leurs informations et leurs ressources, en exploitant les connexions qu'ils ont entre eux. Ces attaques distribuées, nécessitant plusieurs nœuds répartis à différents endroits dans le réseau, sont généralement plus évoluées et plus dangereuses. Par ailleurs, en raison de l'intervention de plusieurs nœuds intermédiaires, leurs détection et l'identification précise de leurs origines sont plus complexes.

3.6.3 Menace à propres aux réseaux ad-hoc

Suite aux diverses caractéristiques, ces réseaux sont vulnérables à plusieurs attaques [Wu et al., 2007; Kannhavong et al., 2007; Singh and Kaur, 2013]. Chaque attaque a son impact sur un point de fonctionnement ou sur un protocole dans une couche précise du modèle OSI. Cependant, il existe d'autre type d'attaque dite intelligente ou l'attaquant affecte plus que une couche.

Attaque liée à la couche physique

Les attaques sur la couche physique sont orientées vers le matériel, elles utilisent les ressources matérielles pour obtenir un effet. Ces attaques sont simples à exécuter par rapport à d'autres attaques. Elles ne nécessitent pas une connaissance complète de la technologie en citant certaines de ces attaques (l'espionnage, brouillage, etc ...) :

La surveillance ou l'espionnage : L'écoute peut également être définie comme l'interception et la lecture des messages et des conversations par les attaquants. Toute communication sans fil peut facilement être interceptée par un récepteur qui entend sur la fréquence correcte. L'objectif principal de ces attaques est d'obtenir des informations confidentielles qui doivent être gardées secrets pendant la communication. Les informations peuvent contenir par exemple la clé privée, clé publique, l'emplacement ou les mots de passe.

Brouillage : Brouillage est une classe spéciale d'attaques DoS qui est initiée par le nœud malveillant après avoir déterminer la fréquence des communications. Dans ce type d'attaque, le brouilleur émet des signaux ainsi que des menaces de sécurité, cette attaque empêche également la réception de paquets légitimes.

Interférence : c'est une intervention active et une attaque par déni de service qui bloque le canal de communication sans fil, ou distorse la communication. Les effets de telles attaques dépendent de leurs durées, et leurs protocoles de couches supérieures. L'attaquant peut modifier l'ordre des messages ou d'essayer de répéter les anciens messages.

Attaque liée à la couche Liaison de donnée

Les algorithmes utilisés dans la couche liaison de données sont sensibles à nombreuses attaques DoS. Les attaques de couche MAC peuvent être classées en fonction de l'effet qui touche l'état du réseau et ses ensembles des nœuds. Les effets peuvent être mesurés en termes d'échec de découverte de route, la consommation d'énergie, la rupture de lien, l'initiation de découverte de route. Le mauvais comportement d'un nœud peut être simplement dans l'intérêt égoïste ou avec des intentions malveillantes.

Attaque liée à la couche Réseaux

Les attaques de la couche réseau ont pour objectif de détourner des flux afin d'écouter, d'analyser ou de perturber le processus de routage ce qui se traduit généralement par un déni de service, on peut les classer comme suit :

Comportement égoïstes des nœuds : cette catégorie affecte directement les performances des nœuds et n'interfère pas avec le fonctionnement du réseau. Il peut comprendre deux facteurs importants :

– Conservation des ressources (batterie, mémoire)

– Gagner par inéquitable de la bande passante

Les nœuds égoïstes peuvent refuser de participer au processus de transfert ou rediriger les paquets intentionnellement dans le but de conserver les ressources. Ces attaquants exploitent le protocole de routage à leurs propres avantages. La suppression des paquets est l'un des principales attaques par nœud égoïste qui conduit à une congestion dans le réseau. La plupart des protocoles de routage n'ont pas de mécanisme pour détecter si des paquets sont transmis ont été reçu ou non à l'exception DSR(Dynamic source routing).

Comportement malveillant des nœuds : Le rôle principal du nœud malveillant est de perturber le fonctionnement normal du protocole de routage. L'impact de ces attaques est augmenté lorsque la communication a eu lieu entre les nœuds voisins. Les attaques de ce type sont résumées dans les catégories suivantes.

– Déni de service (DoS) : Ces types de menaces produisent une action malveillante à l'aide des nœuds compromis qui forment des risques de sécurité grave. En présence des nœuds compromis, il est très difficile de détecter le routage compromis. La route compromis apparait comme un chemin normal, mais elle conduit à de graves problèmes. Par exemple, un nœud compromis pourrait participer à la communication, mais il supprime les paquets cela conduit à la dégradation de la qualité de service offert par le réseau.
– Atteintes à l'intégrité du réseau : L'intégrité du réseau est une question importante, afin de fournir une communication sécurisée et de qualité de service dans le réseau. Il y a autant de menaces qui exploitent le protocole de routage pour introduire de fausses informations de routage.

Attaque liée à la couche Transport

Vol de session : les Attaquants dans le détournement de session exploite les sessions non protégées après leurs installations initiales. Dans cette attaque, l'attaquant usurpe l'adresse IP du nœud victime et trouve le numéro de séquence correct c'est à dire prévu par la destination, ensuite lance un ensemble des attaques DoS.

Dans le détournement de session, le nœud malveillant essaye de recueillir des données sécurisées telles que mots de passe, des clés secrètes, noms des session et d'autres informations à partir des nœuds victimes.

Les détournements de sessions sont également connus comme l'attaque qui a d'incidence sur le protocole OLSR (Optimized Link State Routing Protocol). Le problème du protocole TCP se produit lorsqu'un nœud malveillant lance une attaque pour détourner une session TCP ouverte.

Dans la figure 3.15, L'attaquant injecte des données de session du nœud "A", il envoie un paquet ACK au nœud "B". Le paquet ne contient aucun numéro de séquence qui est attendu par le nœud B, lorsque le nœud "B" reçoit le paquet essaye de resynchroniser la session TCP avec le nœud "A". Ce processus est répété plusieurs fois ce qui conduit à une tempête de ACK "ACK storm".

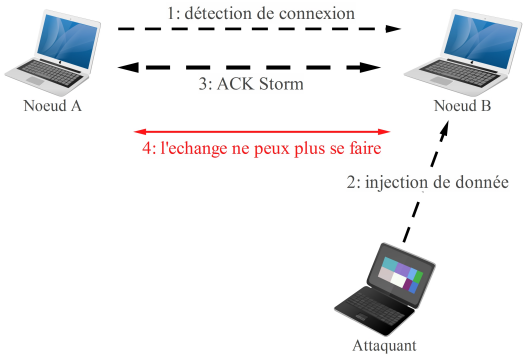

Figure 3.15 – Vol de session

Attaque SYN Flooding : Les attaques d'inondations de SYN sont des attaques par déni de service (DoS) qui consiste à consommer toutes les ressources de la couche transport TCP d'un nœud, dans les quelles l'attaquant crée un grand nombre de demi-connexion TCP ouverte avec le nœud victime. Ces demi-connexions ouvertes ne vont jamais compléter l'établissement de session pour ouvrir complètement la connexion.

Attaque liée à la couche Application :

Les protocoles de la couche application sont également vulnérables à des nombreuses attaques DoS. La couche d'application contient des données d'utilisateur, elle supporte les protocoles tels que HTTP, SMTP, FTP et TALNET, qui offre des nombreuses vulnérabilités et des points d'accès pour les attaquants.

3.7 Outils de la sécurité

Dans la littérature il existe plusieurs outils de sécurité d'information et dans cette partie nous décrivons quelques techniques [Galice, 2007] :

3.7.1 Les algorithmes cryptographiques

La cryptographie c'est la science de sécurisation de l'information, elle offre une protection aux données sensibles pour les transmettre d'une façon protégée sur des réseaux ouverts tel que internet. Il existe deux catégories d'algorithmes de chiffrement :

- Les algorithmes de chiffrement symétrique utilisent la même méthode et la même clé pour chiffrer et déchiffrer un message. La clé secrète doit être connue au préalable par les communicants. L'un des problèmes de cette technique c'est la clé, qui doit rester totalement confidentielle entre les communicants et doit être aussi transmise au correspondant de façon sécurisée. La mise en œuvre d'un chiffrement symétrique peut s'avérer difficile, surtout avec un grand nombre de correspondants car il faut autant de clés que de correspondants.
- Les algorithmes de chiffrement asymétrique dit aussi à clé publique nécessitant l'intervention de deux clés, une clé publique qui est partagée par tous les nœuds qui vont chiffrer les données avant de les envoyer, et une clé privée maintenue secrète chez le nœud récepteur pour déchiffrer les données chiffrées avec la clé publique. Il faut s'assurer que la clé privée correspondante ne peut pas être trouver à partir de la clé publique correspondante. L'algorithme RSA, introduit par [Rivest et al., 1978] est un algorithme de chiffrement asymétrique.

3.7.2 Les certificats électroniques

Un certificat est un élément numérique qui est utilisé pour identifier et prouver l'identité d'une entité et aussi qu'il est associé à une clé publique. Les certificats sont signés et transmis de façon sécuritaire par un tiers de confiance appelé autorité de certification (*Certificate Authority* ou CA).

L'autorité de certification c'est l'entité de confiance chargée de délivrer les certificats, de leur attribuer une date de validité, ainsi que de révoquer éventuellement des certificats avant cette date en cas de compromission de la clé ou s'il y a un changement au niveau du détenteur.

la structure des certificats est normalisée par le standard $X.509$ de l'UIT [Housley et al., 2002] et *OpenPGP* [Callas et al., 2007], qui définissent les informations contenues dans le certificat :

- La version.
- Le numéro de série de l'autorité de certification.
- L'algorithme de signature du certificat.
- Le nom de l'autorité de certification.

- Le nom du propriétaire du certificat.
- La date de validité du certificat.
- Le propriétaire du certificat.
- La clé publique du propriétaire.

L'ensemble de ces informations est signé par l'autorité de certification. Cela signifie qu'une fonction de hachage crée une empreinte de ces informations (hach), puis ce hach est chiffré à l'aide de la clé privé de *CA*. La différence entre les formats d'un certificat X.509 et un certificat OpenPGP est qu'un certificat X.509 ne peut contenir qu'un seul identifiant, et ne peut être signé que par une seule CA. Un certificat OpenPGP peut contenir plusieurs identifiants, et peut être signé par une multitude d'autres CA.

3.7.3 Les Fonctions de hachage

Une fonction de hachage [Rogaway and Shrimpton, 2004] est une fonction à sens unique qui permet d'obtenir une empreinte appelée aussi signature(condensé), de longueur fixe à partir de texte de longueur variable finie. La fonction de hachage doit associe un seul condensé à un texte clair. Cela signifie que la moindre modification du texte entraine la modification de son empreinte. Cette fonction doit être facilement calculable et elle doit assurer qu'il est impossible de retrouver le message original à partir de l'empreinte. En expédiant le message accompagné à son haché il est possible de garanti l'intégrité d'un message. C'est-à-dire que le destinataire peut vérifier que le message n'a pas été altéré durant la communication.

Une fonction de hachage est dite cryptographique si elle assure les conditions suivantes :

1. résiste à l'attaque sur la première pré-image : Il est très difficile de trouver le message à partir de la valeur de hachage.
2. résiste à l'attaque sur la seconde pré-image : A partir d'un message donné et de sa valeur de hachage, il est très difficile de générer un autre message qui donne la même valeur de hachage.
3. résistance aux collisions : Il est très difficile de trouver deux messages aléatoires qui donnent la même valeur de hachage.

3.7.4 Les signatures numériques

La signature numérique [Ghosh and Datta, 2012] est définie comme des données ajoutées à un message, ou une transformation cryptographique d'un message, permettant à un destinataire de :

1. Authentifier l'auteur d'un document électronique.
2. Garantir son intégrité.
3. Protéger contre la contrefaçon (seul l'expéditeur doit être capable de générer la signature), assurer alors la non-répudiation.

La signature électronique est basée sur l'utilisation conjointe d'une fonction de hachage, et de la cryptographie asymétrique. La signature numérique comprend deux étapes :

a) Évaluation de l'empreinte de message : l'émetteur commence par générer une empreinte, qui est une représentation réduite et unique du message complet, à l'aide d'une fonction de hachage.

b) Signature du condensé : l'émetteur chiffre cette empreinte avec un algorithme asymétrique à l'aide de sa clé privée. Il obtient une signature électronique qu'il appose au message original avant d'émettre l'ensemble, message et signature, sur le réseau.

3.7.5 Les infrastructures de gestion de clé publique

Une infrastructure à clés publiques ou *Public Key Infrastructure (PKI)* est un système destinée à gérer les certificats électroniques. Il organise la publication, la gestion et la distribution des clés utilisées dans la cryptographie à clé publique. Elle permet de garantir une relation unique entre l'identité et la clé publique et assure tous les objectifs de sécurité. La sécurité d'une telle infrastructure est assurée par les standards et les logiciels qui réglementent l'utilisation des certificats et les paires de clés publiques/privées. Leur fonctionnement techniquement basé sur les algorithmes de chiffrement symétrique et asymétrique, ainsi que les schémas de signature numérique. Une PKI se compose d'une autorité de certification (CA), d'une autorité d'enregistrement *Register Authority (RA)* et d'un service de publication. Ces autorités sont en charge de plusieurs services dans la gestion des certificats (génération, validation, révocation et publication), la gestion des usagers, la définition et l'utilisation des politiques de confiance, et consultent d'autres services du réseau pour définir toutes les instances nécessaires à un certificat comme un service de nom *(X.509)*, un service d'horodatage pour les dates et les durées de validité.

3.8 Les Attaques de la Couche Réseau

La fonction de routage est une technique essentielle pour l'acheminement des données. Dans les réseaux ad hoc tous les nœuds ayant le même niveau d'autorité et

de sécurité, la chose qui les rendent vulnérables aux plusieurs catégories d'attaque. Malheureusement, cette collaboration entre les nœuds a été détournée par certain nœud malveillant pour attaquer le réseau.

Dans cette section, nous verrons dans la première partie quelques attaques exploitables au niveau des différentes étapes des protocoles de routage dans les réseaux ad hoc. Ensuite, nous présentons dans la deuxième partie l'attaque de type *black hole*, celles-ci ayant un impact important sur nos travaux.

Attaque par Inondation (FLOODING ATTACK)

L'objectif de l'attaque des inondations [Yi et al., 2005] est d'épuiser les ressources du réseau, tel que la bande passante et de consommer des ressources d'un nœud de calcul, et la puissance de la batterie pour interrompre l'opération d'acheminement et provoquer une forte dégradation dans les performances du réseau. Par exemple, dans le protocole *AODV*, un nœud malveillant peut diffuser un grand nombre de *RREQ* dans une courte période à un nœud de destination qui n'existe pas dans le réseau. Car aucune unité ne répondra aux *RREQ*, ces *RREQ* vont propager dans l'ensemble du réseau. En conséquence, toute la puissance de la batterie des nœuds, ainsi que la bande passante de réseau sera consommée et pourrait conduire à un déni de service. Dans [Desilva and Boppana, 2005] ils ont montré qu'une attaque d'inondation peut diminuer le débit jusqu'à 84 %.

Attaque par Rejeu (REPLAY ATTACK)

Dans un MANET, la topologie change fréquemment en raison de la mobilité des nœuds. Cela signifie que la topologie du réseau actuelle peut ne pas exister dans le futur. Dans une *replay attack* [Zhen and Srinivas, 2003] , un nœud enregistrent les messages du contrôle valide d'un autre nœud et les renvoie plus tard. Cela provoque que d'autres nœuds enregistre dans leur table de routage des routes périmées. L'attaque par rejeu peuvent être détournée pour usurper l'identité d'un nœud spécifique ou tout simplement pour perturber le fonctionnement de routage dans un MANET avec des informations expirées.

L'attaque de trou de ver (WORMHOLE ATTACK)

L'attaque de trou de ver [Hu et al., 2006] est une des attaques les plus sophistiquées et sévères dans les MANET. Dans cette attaque, une paire d'attaquants enregistrent les paquets à un endroit et les reproduit à un autre endroit en utilisant un réseau privé à une vitesse élevée. La gravité de cette attaque est qu'elle

peut être lancée contre toutes les communications qui offrent l'authenticité et la confidentialité.

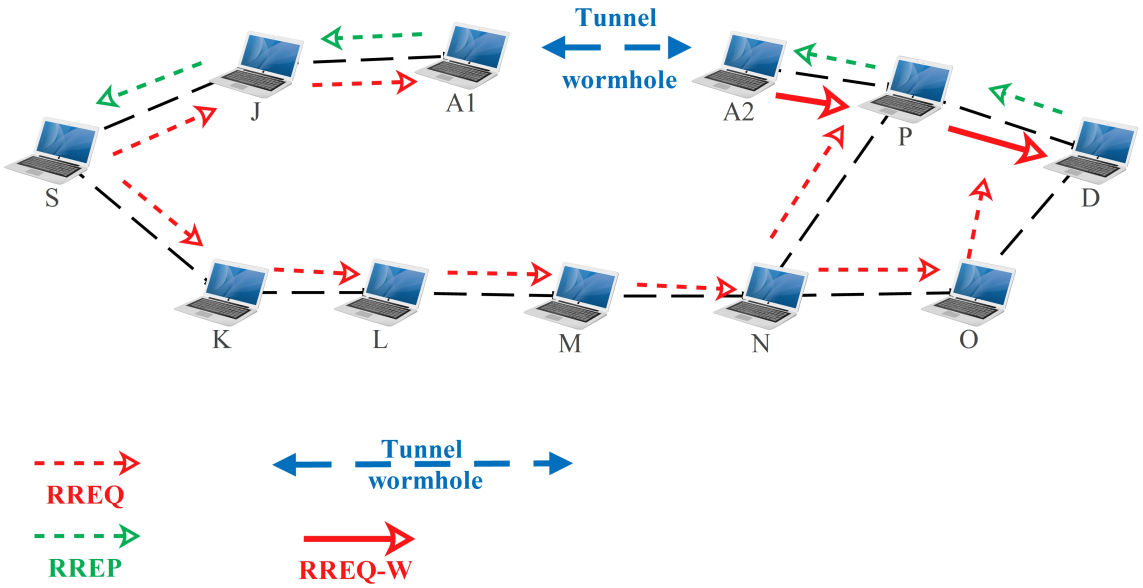

Figure 3.16 – l'attaque de trou de ver.

Un exemple de l'attaque de trou de ver dans un protocole de routage réactif. Dans la (figure3.16) nous supposons que les nœuds A1 et A2 sont deux attaquants et le nœud S est la cible d'être attaquée. Pendant l'attaque, lorsque le nœud source S émis un *RREQ* pour trouver une route vers un nœud de destination D, ses voisins J et K reçoivent le paquet *RREQ*. Cependant, le nœud A1, qui a reçu le *RREQ* va le retransmettre à travers les tunnels à son partenaire A2. Puis, le nœud A2 rediffuse ce *RREQ* à son voisin P. Ce *RREQ* atteindra rapidement le nœud D en premier temps par rapport aux autres chemins. Par conséquent, le nœud D choisira l'itinéraire $D-P-A1-A2-J-S$ et unicast une *RREP* vers le nœud source S et ignore les autres *RREQ* même s'ils arrivent plus tard. Par conséquent, S sélectionne l'itinéraire $S-J-A1-A2-P-D$ qu'en effet traverse A1 et A2 pour envoyer ses données.

L'attaque de trou noire (BLACK HOLE ATTACK)

Le trou noir [Kurosawa et al., 2007] est un type d'attaque qui peut être facilement utilisée dans le processus de routage dans les réseaux ad-hoc. Elle apparait quand un nœud refuse de coopérer dans le processus de routage. En effet, les nœuds égoïstes refusent de relayer les messages de contrôle des autres nœuds et qui sont destinés à être diffusés dans le réseau entier, Dans cette attaque :

- Le nœud malveillant détecte l'existence d'une route active et prend note de l'adresse de la destination.
- Le nœud malveillant prépare un paquet route réponse (*RREP*) dans lequel : le champ d'adresse de destination est défini sur l'adresse de destination falsifiée, le numéro de séquence est fixé à une plus grande valeur et le nombre de saut est mis à une petite valeur.
- Le nœud malveillant envoi cette réponse *RREP* vers le plus proche nœud intermédiaire appartenant à la voie active réelle (pas nécessairement vers le nœud source lui-même).
- La réponse de route *RREP* reçu par le nœud intermédiaire sera retransmise au moyen de chemin inverse pré-établi vers le nœud source de données.
- Le nœud source mis à jour sa table de routage par les nouvelles informations reçues dans la réponse de route.
- La source utilise la nouvelle route à l'envoi de données.
- Le nœud malveillant commence à ignorer les données de la route à laquelle il appartient.

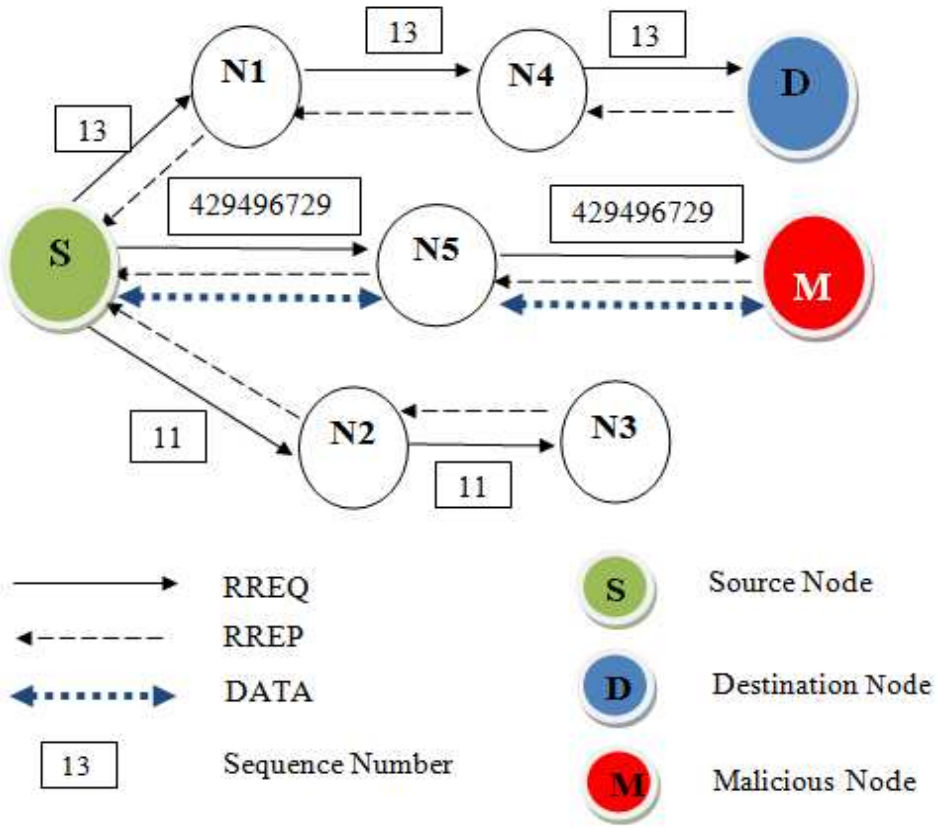

Figure 3.17 – l'attaque black hole

3.9 Conclusion

Nous avons présenté dans ce chapitre la problématique de la sécurité du protocole routage dans les réseaux ad hoc qui n'a pas été entamée lors des premières propositions de protocole de routage. Le routage est une des fonctions de base essentielles au bon fonctionnement des réseaux ad hoc. Il s'agit d'un défit qui a fait l'objet de très nombreuses travaux de recherches car il n'existe pas des solutions au préalable sécurisée, cela a donné lieu à de nouvelles techniques qui tentent de résoudre ce problème.

Par la suite, on a donné une description sur les trois catégories globale du protocoles de routage selon le groupe MANET (proactifs, réactifs ou hybrides). Dans la fin nous avons abordé quelques attaques au niveau des étapes des protocoles de routage dans les réseaux ad hoc. Ensuite, nous avons présenté l'attaque de type *black hole*, celle-ci ayant un impact important sur nos travaux.

Les réseaux Ad-hoc sont vulnérables aux plusieurs types d'attaques. D'après l'étude que nous avons faite dans cette première partie, les réseaux Ad-hoc se caractérisent par des qualités qui les distinguent par rapport aux réseaux filaires. Certaines de ces qualités peuvent représenter des avantages dans des cas, comme ils peuvent constituer un handicap dans d'autres situations.

De ce fait, les réseaux Ad-hoc présentent des challenges difficiles dans la sécurisation des protocoles. Il faut non seulement éviter les nombreuses attaques causées par les attaquants internes ou externes et les nœuds compromis et malveillants, mais aussi veiller à ce que la dégradation des performances causée par les mécanismes de sécurité soit limitée.

Dans le chapitre suivant, nous allons entamer les approches existantes dans la littérature pour fournir des solutions contre les attaques dans les réseaux ad hoc.

Chapitre 4

Mécanisme de détection et élimination de nœud black hole dans AODV

Contents

4.1 Introduction

Les services de sécurité sont essentiels pour les MANETs, en prenant le cas des applications militaires pour assurer un déploiement à grand échelle. Beaucoup de solutions de sécurité ont été employé dans la couche réseau soit comme protocoles de routages sécurisés ou comme des couches et des techniques introduisant la notion de sécurité pour des protocoles déjà existant.

Les protocoles de routage dans les Manet comptent sur la coopération entre les nœuds en raison de l'absence d'une administration centralisée et d'un support fixe. D'autre part, l'hypothèse que tous les nœuds ont le même niveau élevé de confiance rend ce réseau facile à attaquer par des nœuds malveillants, soit pour perturber le fonctionnement de routage comme le déni de service (DOS) ou pour refuser la participation dans le routage (nœud égoïste). Beaucoup de techniques ont été proposées pour lutter contre ces attaques, qui sont classées en cinq catégories principales.

Dans ce chapitre nous allons présenter notre première contribution qui focalise sur les techniques utilisant des méthodes de détection des attaques de trou noir *black hole*. Notre proposition est basée sur le concept de communication inter-couche entre la couche réseau et la couche MAC.

Nous allons tout d'abord commençer par une étude descriptive sur les travaux de recherche qui ont été déjà proposés dans la littérature, nous allons les classé en catégories dans la section 4.2. Ensuite, dans la section 4.3 nous allons décrire l'aspect d'architecture de communication inter-couche. La section 4.4 présentera notre contribution CrossAODV qui permet de détecter et éliminer les nœuds *black hole*. Nous évaluerons l'efficacité de CrossAODV dans la section 4.5 puis nous analyserons ces résultats en utilisant des paramètres d'évaluation des performances dans la section 4.6. Finalement la section 4.7 conclura cette partie.

4.2 Positionnement bibliographique

Plusieurs travaux dans la littérature proposent des solutions au problème de la sécurité dans les réseaux mobiles Ad Hoc. Afin de catégoriser ces travaux nous avons fait une étude étendue sur les différentes propositions.

le nœud attaquant trou noir bénéfice aux différents tâches des protocoles de routage pour nuire à leurs bon fonctionnement. En général un attaquant dans le protocole AODV change l'information qui existe dans les paquets de contrôle pour détourner tous les flux de données ver lui.

Les méthodes de détections des attaques peuvent être classées en cinq catégories

générales [Jain and Khunteta, 2015] : Techniques de détection basées sur le numéro de séquence, sur la cryptographie, sur le point de vue des nœuds voisins, sur les systèmes de détection d'intrusion (IDS), et d'autres basées sur la confiance.

4.2.1 Techniques de détection basées sur le numéro de séquence

Plusieurs travaux ont proposé des techniques pour détecter l'attaquant *black hole* à base de numéro de séquence. La majorité des approches de cette catégorie permettant d'étudier la valeur de numéro de séquence de destination *(NSD)*. Les auteurs dans [Jaiswal and Kumar, 2012] proposent une méthode qui permet à la station source de collecter les paquets RREP, ensuite comparer leurs numéros de séquence. Si le premier paquet RREP à une valeur de numéro de séquence de destination grande, il sera ignoré. La décision finale pour la sélection du chemin de communication se fait à base du reste des paquets collectés.

Dans [Harmandeep and Manpreet, 2013], les auteurs ont crée une table qui contient le couple numéro de séquence de destination des paquets RREP reçu et l'identité de l'émetteur. Le nœud source va comparer les entrées de cette table avec numéros de séquence dans RREQ, l'entrée avec le grand numéro de séquence sera supprimée.

[Lalit et al., 2011] ils ont proposé une méthode pour trouver les routes sécurisées et prévenir les nœuds malveillants dans les MANETs en vérifiant s'il existe une différence importante entre le numéro de séquence du nœud source et le nœud intermédiaire qui a envoyé la première RREP. En règle générale, la première réponse sera du nœud malveillant avec un numéro de séquence de destination très élevé, la réponse sera stockée comme la première entrée dans *RR-table* (route replies table). Ensuite, le nœud source va comparer le premier numéro de séquence de destination reçu avec son numéro de séquence, s'il existe une grande différence entre eux, certainement cette réponse vient du nœud malveillant, par conséquent il va supprimer cette entrée de la table. La méthode proposée offre les avantages suivants :

1. Le nœud malveillant est identifié dans la phase initiale et il est retiré immédiatement.
2. Aucune modification n'est faite dans les autres opérations du protocole AODV.
3. Une meilleure performance avec une légère modification.

[Kamarularifin et al., 2011] ont proposé un ERDA (Enhance Route Discovery for AODV), la source écrase l'ancien paquet RREP par le nouveau à travers la comparaison des numéros de séquence destination du paquet RREP reçu. Les paquets

RREP qui ont été capturé sont enregistré dans une table appelé rrep_table.

[Jhaveri et al., 2012] propose de comparer la valeur de numéro de séquence de destination reçu dans un RREP avec une valeur maximale. Si la valeur du numéro de séquence reçu est grande par rapport à la valeur fixée le paquet sera ignoré. La valeur maximale est calculée dans un intervalle de temps suit une combinaison de trois valeur (numéro de séquence de RREP, numéro de séquence dans la table de routage ainsi le nombre de réponse reçu durant cet intervalle).

[Vani and Sreenivasa Rao, 2011] Ont proposé une solution pour le problème d'attaque trou noir qui est la comparaison du numéro de séquence le plus élevé avec la valeur du seuil. Dans AODV, quand le nœud source reçoit un paquet RREP, il vérifie d'abord la valeur du numéro de séquence dans sa table de routage, la source accepte le paquet RREP (Route Reply) si le numéro de séquence est supérieur au numéro de séquence qui est dans la table de routage. Sa solution ajoute une valeur de seuil pour comparer le numéro de séquence de RREP le plus élevé dans chaque intervalle de temps. Si la valeur de *RREP-seq-no* est plus élevée que la valeur du seuil, Le nœud est soupçonné d'être malveillant et il va être ajouté à la liste noire et rejeter toutes les messages de réponse arrivant à partir de ce nœud. Dans la simulation, si le RREQ est reçu par un nœud trou noir, Il va générer RREP avec :

1. Numéro de séquence le plus élevé = 1000
2. Numéro de séquence différence = 50

[Tan and Keecheon, 2013] ont défini plusieurs valeurs de seuil pour chaque simulation ensuite, ils comparent le numéro de séquence de destination avec la valeur de seuil si cette valeur est grande dans ce cas le paquet RREP est ignoré. Dans [Azza et al., 2014] ont faisait des prédictions avant de mettre à jour la valeur de seuil.

Les auteurs dans [Raj and Swadas, 2009] ont modifié le comportement d'AODV pour inclure un système d'apprentissage permettant de vérifier le numéro de séquence de RREP reçu. Quand le nœud source reçoit un paquet RREP il compare le numéro de séquence du RREP reçu avec une valeur de seuil.

Le nœud répondant est soupçonné d'être un trou noir si son numéro de séquence est supérieur à la valeur de seuil. Le nœud source ajoute le nœud suspect à sa liste noire, et propage un message de contrôle appelé une alarme pour mettre à jour la liste noire de ses voisins. Le seuil est la moyenne calculée de la différence entre le numéro de séquence de destination dans la table de routage et le numéro de séquence de destination dans le RREP dans une période de temps. L'avantage principal de cette méthode est que le nœud source annonce le trou noir à ses voisins afin de l'isoler.

La méthode proposée dans [Yerneni and Sarje, 2012] se compose de deux parties : la phase de suspicion et la phase de confirmation. Quand un nœud reçoit plusieurs RREP il retarde le traitement du RREP et lance la phase 1.

1. Phase de suspicion : dans cette phase le nœud classe les paquets RREP suivant les numéros de séquence de façons descendante et fait une comparaison avec la moyenne des autres RREP, si l'une des valeurs et supérieure à la moyenne et le délai de réponse et minimum dans ce cas le nœud qui a envoyé ce RREP est suspecté malicieux. (voir figure 4.1)
2. Phase de confirmation : quand un nœud est suspecté, la station source prépare un nouveau paquet MREQ qui contient un nombre aléatoire prédéterminé entre la station source et la destination en préalable et l'envoie sur tous les chemins identifiés. Chaque chemin contient un nombre aléatoire différent de l'autre. Quand la station destination reçoit MREQ (voir figure 4.2), elle répond par un paquet MREP qui contient le même nombre aléatoire définit par la source, si ce dernier reçoit plusieurs MREP avec le même nombre aléatoire cela rend la destination confiante et choisit l'un des chemins avec un numéro de séquence plus élevé. Le nœud malveillant n'a aucune connaissance sur le nombre aléatoire choisi par la destination et la probabilité pour choisir la même valeur petite et s'il a choisi une valeur elle doit être différente par rapport à la valeur de la station source.

```
Receive REQUEST:
    {
      Source node:
                    Discard.

       Intermediate node:
                    If fresh path is there towards
destination
                     then
                              Send REPLY;
                    Else
                              Forward REQUEST;
       Destination node:
                    Prepare REPLY and unicast it to
source.
      }

    Receive REPLY
    {
      Source node:
          If REPLY sent by suspected node then
          {
               Select random number;
               Send MREQUEST;
          }
          Else
               Send Application data if the path is fresh;
        Intermediate node:
               Forward REPLY;
      }
```

FIGURE 4.1 – Phase de suspicion [Yerneni and Sarje, 2012]

```
    Receive MREQUEST
     {
       Destination/intermediate node for which
        MREQ is intended:
          {
               Select a random number corresponding
                to source  node;
               Prepare MREPLY and unicast to source
               node;
           }
      }


     Receive MREPLY
      {
          Source node:
             Wait until it receives same number via
          different    paths and send data through fresh
path;
       }
```

FIGURE 4.2 – Phase de confirmation [Yerneni and Sarje, 2012]

[Subash and Surya, 2011] Ont proposé un algorithme simple qui n'affecte pas le fonctionnement du protocole. Le protocole AODV utilise un pré-traitement appelé Pre_Process_RREP. Le processus continu d'accepter les paquets RREP et fait appel à une procédure Compare_Pkts (p1 paquet, p2 paquet) qui compare effectivement les numéros de séquence de destination des deux paquets et sélectionne le paquet avec le numéro de séquence de destination d'ordre supérieur si la différence entre les deux nombres est élevée. Le paquet contenant le numéro de séquence de destination exceptionnellement élevé est soupçonné d'être transmis par un nœud malveillant. Dans ce cas un message d'alerte contenant l'identifiant du nœud est diffusé aux nœuds voisins de telle sorte que tout message reçu à partir du nœud malveillant sera rejeté. Une liste des nœuds malveillants doit être maintenue par les nœuds qui participent à la communication, ce qui permet de prévenir contre une attaque trou noir.

Dans [Nital et al., 2010], les auteurs ont publiés une solution sous le titre "Improving AODV Protocol against Blackhole Attacks "Ils ont proposé un algorithme simple qui n'affecte pas le fonctionnement du protocole AODV, mais ont utilisé une procédure de pré-traitement appelée *Pre_ReceiveReply*(Packet P) (voir algorithme 1).

Algorithm 1 Pseudo-code for Pre_ReceiveReply (Packet P)

```
Pre_ReceiveReply (Packet P)
t0 = get(current time value)
t1 = t0 + MOS_WAIT_TIME
while CURRENT_TIME <= t1 do
    Store P.Dest_Seq_No and P.NODE_ID In Cmg_RREP_Tab table
end while
while Cmg_RREP_Tab is not empty do
    Select Dest_Seq_No from table
  if Dest_Seq_No >>>= Src_Seq_No then
      Mali_Node = Node_Id
      discard entry from table
  end if
    select Packet q for Node_Id having highest value of Dest_Seq_No
    ReceiveReply(Packet q)
end while
```

4.2.2 Technique de détection basée sur la cryptographie

Plusieurs travaux ont proposé pour sécuriser le protocole AODV contre l'attaque *black hole*. Il utilisent des primitives cryptographique symétrique ou asymétrique

Table 4.1 – Tableau de Comparaison des approches basées sur le NS

Méthode	Technique	Modification	fonction modifiée	taux de contrôle	délai	paquet +
[Jaiswal and Kumar, 2012]	différence entre NSD et NSS	oui	RREP	non	non	-
[Harmandeep and Manpreet, 2013]	différence entre NSD et NSS	oui	RREP	non	oui	-
[Lalit et al., 2011]	supprime premier paquet	non	-	non	non	-
[Kamarularifin et al., 2011]	comparaison entre NSD et NSS	oui	RREP	oui	non	-
[Jhaveri et al., 2012]	comparaison avec une valeur maximale	non	RREP	non	non	-
[Vani and Sreenivasa Rao, 2011]	comparaison avec seuil fixe	non	RREP	oui	non	-
[Tan and Keecheon, 2013]	environnement basé sur un schéma de détection avec seuil	oui	RREP RREQ	non	non	-
[Raj and Swadas, 2009]	comparaison avec un seuil dynamique	oui	RREP	oui	oui	Alert
[Azza et al., 2014]	un seuil dynamique	oui	RREP	non	oui	-
[Yerneni and Sarje, 2012]	suspicion et confirmation	oui	RREP	oui	oui	MREQ MREP
[Subash and Surya, 2011]	Collecter et Comparer	non	RREP	non	oui	-
[Nital et al., 2010]	Pré-traitement	oui	RREP	non	oui	-

pour assurer l'authentification, ou la signature électronique et les fonctions de hachage pour assurer l'intégrité des informations du routage.

Dans [SB and Benni, 2013],Les auteurs ont utilisé la cryptographie asymétrique avec RSA. Dans cette technique le nœud source chiffre le nombre de saut et le Numéro de séquence Destination avec la clé public du nœud destination, cela rend les deux informations accessibles seulement par destination avec sa clé privée. Par contre [Singh et al., 2012] utilisent l'autre type de cryptographie (symétrique) conjointe avec une fonction de hachage cyclique pour générer un identifiant unique pour chaque nœud.

Pour assurer l'authentification du nœud participant dans le processus de routage [Sowmya and Mayuri, 2013], utilisent la notion de cryptographie à seuil. L'information secrète représente les identités de l'ensemble des nœuds participant dans le chemin vers la destination. Dans cette approche le nœud malveillant ne peut pas intervenir dans le processus de routage car il ne dispose pas les informations de sécurité.

[Sachan and Khilar, 2011] proposent un régime qui utilise un schéma d'authentification du message à base de fonction de hachage (HMAC). Ce dernier fournit une vérification rapide du paquet ainsi que l'authentification du nœud participant dans le routage. Dans ce cas, chaque nœud détient un HMAC et la clé de sécurité est partagée entre la station source et la station destination.

[Arya and Rajput, 2014] tous les nœuds ont la même table des clés partagées, chaque nœud choisit sa clé à base du nombre de saut ce qui fait qu'il est difficile pour un intrus d'intercepter la clé utilisée dans la communication.

4.2.3 Technique de détection basée sur les IDS

[Abdelhaq et al., 2011] Ont proposé un mécanisme de sécurité de routage Local Intrusion Detection (LID), la détection localement de l'attaquant signifie que le nœud intermédiaire suspecté (nœud N5 voir figure 4.3) envoie un RREP (Route Reply) vers le nœud de source (nœud N1) en mode unicast à travers le nœud précédent (nœud N4). C'est le nœud intermédiaire qui effectue le processus de détection et non pas le nœud source. Tout d'abord, le nœud précédent tamponne le paquet RREP, puis, il utilise une nouvelle route vers le prochain nœud (nœud N6) en envoyant un paquet FRREQ (further Request) à lui. Lorsque le nœud intermédiaire reçoit le paquet FRREP, il extrait les informations à partir du paquet FRREP et se comporte conformément aux règles suivantes :

1. Si le nœud suivant (N6) a une route au nœud intermédiaire (N5) et le nœud de destination (N7), le nœud de saut précédent (N4) supprime le FRREP, puis il

Table 4.2 – Tableau de Comparaison des approches basées sur cryptographie

Méthode	Technique	Modification	fonction modifiée	taux de contrôle	délai	paquet +
[SB and Benni, 2013]	cryptographie asymétrique (RSA)	Non	-	Non	Non	-
[Sowmya and Mayuri, 2013]	cryptographie à seuil identification basée sur la cryptographie	Oui	RREQ RREP	Oui	Non	-
[Sachan and Khilar, 2011]	fonction de hachage HMAC	Oui	RREQ RREP	Oui	Non	-
[Singh et al., 2012]	cryptographie symétrique	Non	-	Non	Oui	-
[Arya and Rajput, 2014]	table des clés partagées	Non	-	Non	Non	-

envoie un RREP vers le nœud source.

2. Si le saut suivant (N6) n'a pas de route au nœud de destination (N7) ou le nœud intermédiaire (N5), ou les deux (N5 et N7), le nœud précédent (N4) supprime le RREP tamponnée et le FRREP ainsi, et diffuse en même temps un message d'alerte pour annoncer qu'il n'existe pas de chemin suffisamment sécurisé à la disposition du nœud de destination (N7).

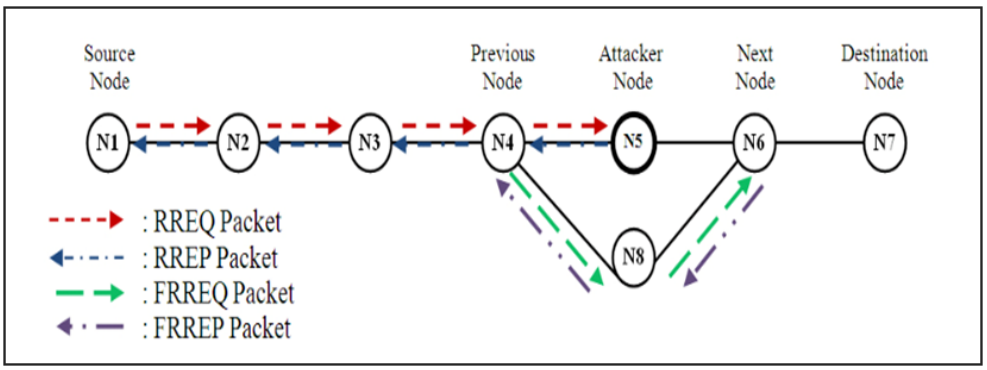

Figure 4.3 – Local Intrusion Détection LID

[Ming-Yang et al., 2010] ont déployé des nœuds IDS dans chaque zone de couverture pour surveiller le réseau, ensuite ils utilisent des fonctions d'estimation pour détecter les nœuds *black hole*. Ces fonctions suspectent un nœud à travers la différence entre le nombre des paquets RREQ et RREP transmis.

[Kamini and Divakar, 2012]propose un système de prévention d'intrusion qui informe l'émetteur de la liste des nœuds malveillants dans le réseau, la prévention se fait à base d'un filtrage de donnée stocké dans une table de profile.

Dans [Sonal, 2013], propose un IDS qui utilise la logique flou pour choisir le prochain nœud dans le chemin. Seulement les nœuds avec une haute priorité peuvent participer à la communication. Une priorité haute est attribuée à un nœud qui à un taux minimal de paquet supprimé et un taux élevé de paquet délivre.

4.2.4 Technique de détection basée sur la confiance ou crédit

[Marchang and Datta, 2012] ont choisi les chemins les plus approuvés par rapport aux chemins courts possèdent une valeur de confiance d'un nœud. La valeur de confiance est calculé à base la moyenne pondéré des valeurs du nœud lui même et les valeurs fournis par les voisins de ce nœud.

[Velloso et al., 2010] ont proposé un protocole d'échange de recommandation, un nœud à la possibilité d'envoyer et de recevoir des recommandations de leurs

Table 4.3 – Tableau de Comparaison des approches basées sur les IDS

Méthode	Technique	Modification	fonction modifiée	taux de contrôle	délai	paquet +
[Abdelhaq et al., 2011]	détection d'intrusions	oui	RREQ RREP	Oui	Oui	FRREQ FRREP
[Ming-Yang et al., 2010]	detection d'anomalies	Non	-	Oui	Non	-
[Kamini and Divakar, 2012]	détection d'anomalies	Non	-	Non	Non	-
[Sonal, 2013]	détection d'intrusion via la logique flou	Non	-	Non	oui	-

voisins. L'établissement d'un chemin de confiance se fait à base d'expérience et de connaissance acquis précédemment sur les recommandations reçus par les voisins.

[Varshney et al., 2014] proposé une approche appelée WAODV (watshdog aodv) qui associe un watshdog à chaque nœud pour assurer que chaque paquet émis par un nœud quelconque sera retransmis par le prochain saut dans le chemin.

[khamayseh et al., 2011] ont ajouté au paquet RREP un champ qui permet de définir le niveau de confiance de chaque nœud. Dans ce cas, la station source choisit le saut suivant qui a la valeur de confiance élevée.

[Khan and Gupta, 2012] créent un vecteur de confiance qui sera calculé dans la phase de découverte de route, ce vecteur représente le degré de confiance de chaque nœud.

4.3 Communication inter-couches

La division en couche a été conçue pour développer des systèmes hétérogènes à grand échelle, elle permet l'interconnexion des systèmes ouverts. Son succès est lié à sa propriété de fournir la modularité et la transparence du système [Shakkottai et al., 2003]. Le modèle en couche utilisé pour les réseaux sans fil n'est pas forcément le plus adapté, notamment à cause de leurs popularités et évolutions des paradigmes de la communication sans fil.

La communication inter-couches (cross-layer) permettent d'atteindre des fins de performance complet, ainsi la facilité d'exploiter des entités précise pour chaque couche. Elle suppose un échange d'information entre des couches, éventuellement non adjacentes.

Architectures inter-couche

Plusieurs types architectures inter-couches sont décrits par [Srivastava and Motani, 2005], permettant d'énumérer les différents modes d'interaction entre les couches, ces types sont illustrés dans la figure 4.4.

1. Architecture ascendante : consiste à remonter les informations des couches inférieures vers les couches supérieures. Généralement, elle est utilisée dans plusieurs cas comme les congestions qui sont notifiées par la couche liaison de donnée.
2. Architecture descendante : consiste à utiliser une communication inverse, des couches supérieures aux couches inférieures. Prennent l'exemple de l'architecture 802.11e du WIFI à travers de la couche supérieure on peut gère l'un des quatre types de la file d'attente *(Best Effort, Background, Video, Voice)*

Table 4.4 – Tableau de Comparaison des approches basées sur confiance

Méthode	Technique	Modification	fonction modifiée	taux de contrôle	délai	paquet +
[Marchang and Datta, 2012]	Valeur de confiance	Non	-	Non	Oui	-
[Velloso et al., 2010]	Système de Recommandation	Non	-	Non	Non	-
[Varshney et al., 2014]	Système de surveillance	Non	-	Non	Non	-
[khamayseh et al., 2011]	système de confiance	Oui	RREP	Non	Oui	-
[Khan and Gupta, 2012]	Vecteur de confiance	Non	-	Non	Oui	-

3. Architecture ascendante descendante : c'est une combinaison du deux types précédents. L'exemple le plus connus c'est le scheduling des paquets entre la couche MAC et la couche supérieure (réseau).
4. Architecture fusionné : consiste à créer une super-couche par la fusion de deux couches. Cette technique est utilisée souvent dans les approches d'optimisations typiquement, le modèle TCP/IP qui combine les couches physique et MAC dans une seule couche d'accès aux réseaux.
5. Architecture adaptative : consiste à développer une couche abstraite suivant les besoins en anticipant le fonctionnement d'une couche inférieure. Ce genre est utilisé dans les cas où il n'y a pas d'accès direct aux couches inférieures.
6. Architectures de calibrage vertical : consiste à adapter une propriété spécifique d'une couche à une autre couche par exemple une architecture permet de conserver les choix en terme de délai en agissant sur l'ensemble des couches.

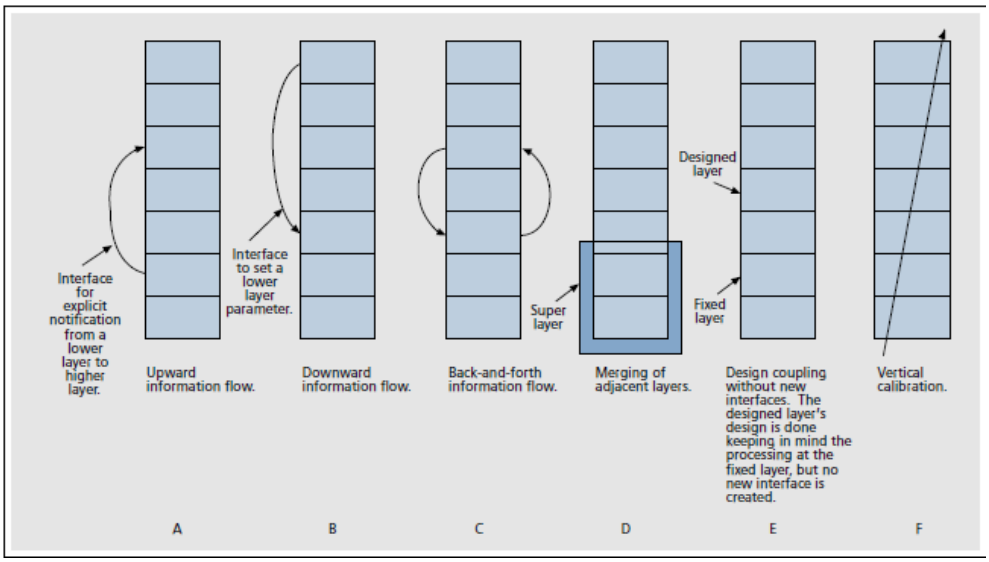

Figure 4.4 – Types d'architectures inter-couche [Srivastava and Motani, 2005]

4.4 Notre contribution

Dans cette partie nous allons détailler notre proposition crossAODV [Azza et al., 2015] qui consiste à introduire une technique dans le protocole de routage AODV, afin de détecter et isoler les nœuds *black hole*. Notre approche est basée sur la communication inter-couches entre la couche accès au support et la couche réseau (voir code B.5). Elle est composée de deux phases principales, la vérification et la validation de route.

La phase de vérification : nous avons apporté des modifications au niveau de la fonction de coordination DCF pour anticiper et obtenir des informations de routage à propos des nœuds voisins. Cette anticipation est réalisée à travers une extension au niveau des deux trames RTS et CTS.

La phase de validation : utilise le mécanisme de communication inter-couches pour prendre une décision sur les chemins obtenus dans la couche réseau via le processus de découverte de route, en se basent sur une jointure avec l'information récupérée par la couche MAC. À ce stade on juge le comportement du nœud voisin. Notre but principal est d'isoler le nœud *black hole* qui affecte à la découverte de route par l'envoi d' une réponse fausse RREP. Après la détection, ce nœud sera ajouté dans une table des nœuds *black hole* notée *black list*.

```
int Mac802_11::command(int argc, const char*const* argv)
{

if (argc == 3) {
//azza cross_layer

if (strcmp(argv[1], "access-aodv") == 0) {
aodv_ = (AODV*) TclObject::lookup(argv[2]);
if (aodv_ == 0)                    return TCL_ERROR;
return TCL_OK;
}
```

Code 4.1 – Fichier mac802-11.cc : création d'un paramètre d'accès en mode inter-couches

4.4.1 Extension de la couche MAC

Le rôle principale de la couche MAC est l'interrogation entre les stations pour accéder au support. Elle assure et gère le contrôle de la transmission des trames. Dans nos simulations, on a utilisé le protocole *IEEE 802.11* du Network simulator 2 (NS-2). Plusieurs types de trames peuvent être reçu par la couche MAC, tels que RTS, CTS, ACK ou des trames DATA. Selon le type de trame reçu, la couche MAC prend sa décision. Le protocole MAC 802.11 suit le dialogue de trames RTS-CTS-DATA-ACK. (voir section 2.3.2) Dans le processus de découverte de route, Le paquet RREQ crée au niveau de la couche routage est transmis vers la couche inférieure, ce paquet sera tamponné pendant que la couche MAC lance le processus d'accès au

médium via la fonction DCF du CSMA/CA. Avant que l'émetteur envoi le paquet RREQ il vérifie si le support est libre puis il crée une trame RTS et le diffuse vers les voisins. Dans notre approche on joint avec cette trame une demande d'information *MAC_RTS_INF* (voir code 4.2) pour savoir l'état de la table du routage du nœud destinataire ou l'intermédiaire et plus précisément sur le chemin ou la destination recherchée Nœud_Destination. Cette phase est appelée la phase de vérification. Le détail est résumé dans l'organigramme 4.5.

```
void Mac802_11::sendRTS(int dst)
{
Packet *p = Packet::alloc();
hdr_cmn* ch = HDR_CMN(p);
struct rts_frame *rf =
        (struct rts_frame*)p->access(hdr_mac::offset_);
...
//azza
hdr_cmn* ch1 = HDR_CMN(pktTx_);
struct hdr_aodv *ah = HDR_AODV(pktTx_);
if (ch1->ptype() == PT_AODV && ah->ah_type == AODVTYPE_RREQ){

rf->rf_fc.fc_wep          = MAC_RTS_INF1;
}
else {
rf->rf_fc.fc_wep          = MAC_RTS_INF0;

}
.....
}
```

Code 4.2 – Fichier mac802-11.cc : jointure de demande d'information avec RTS

À travers l'accès en mode inter-couche le nœud destination vérifie sa propre table de routage, s'il possède un chemin vers la destination ou s'il est lui même destinataire. Il joint avec la trame CTS une information vraie *MAC_CTS_INF* (voir code 4.3) qui permet de dire que ce nœud a une entrée dans sa table de routage vers la destination demandée. À ce moment, il répond par une trame CTS pour dire qu'il est prêt à recevoir le paquet RREQ.

```
void Mac802_11::recvRTS(Packet *p)
{
```

```
struct rts_frame *rf =
(struct rts_frame*)p->access(hdr_mac::offset_);
...
switch(rf->rf_fc.fc_wep) {
case MAC_RTS_INF1:
if (_Cross_layer(ETHER_ADDR(rf->rf_da))){
sendCTS(ETHER_ADDR(rf->rf_ta), rf->rf_duration ,1);
}
break;

default :
sendCTS(ETHER_ADDR(rf->rf_ta), rf->rf_duration ,0);
}
...
}
```

Code 4.3 – Fichier mac802-11.cc : jointure d'information de réponse avec CTS

L'information obtenue via la trame CTS dans chaque étape de vérification, est enregistrée par le nœud source dans une table qui contient : Nœud_Destination, Nœud_Suivant et AODV_CROSS qui sera utilisé par la suite dans la communication inter-couches. Cette table doit être accessible par la suite dans la phase de validation de route. Une variable appelée *AODV_CROSS* permet d'enregistrer l'information obtenue par *MAC_CTS_INF*. Á chaque nouvelle opération de découverte de route cette table doit être réinitialisée. (voir algorithme 2)

Algorithm 2 Pseudo Algorithm of MAC extension

Declaration
MAC_RTS_INF {request information}
MAC_CTS_INF {response information}
$CROSS$ {cross_layer information}
BEGIN
Create RTS packets
$MAC\_RTS\_INF == TRUE$ {Request for routing information added to the RTS}
Send RTS to Neighbor Node
Receive RTS
if ($MAC\_RTS\_INF ==$ **true**) **then**
  if ($ROUTING\_INF ==$ **true**) **then**
    {cross layer check the table of routing }
    $MAC\_CTS\_INF =$ **true** {Prepare CTS with routing information}
  else
    $MAC\_CTS\_INF =$ **false** {there is no routing information}
  end if
end if
Send CTS
Receive CTS with MAC_CTS_INF {routing information }
if ($MAC\_CTS\_INF == TRUE$) **then**
  $AODV\_CROSS = TRUE$
else
  $AODV\_CROSS = FALSE$
end if

L'attaquant *black hole* n'a aucune information de routage. Quand il reçoit une trame RTS, il ne jointe aucune information de vérification. Donc le champ MAC_CTS_INF est vide, ce que fait à la réception de cette trame la variable $AODV\_CROSS$ prend la valeur fausse. (voir algo 4.4.1)

Algorithm 3 Pseudo code for Malicious Node :

1: *BEGIN*
2: *Receive RTS*
3: $MAC\_CTS\_INF =$ **false** {Prepare CTS without routing information}
4: *Send CTS without routing information*
5: *Receive CTS with* MAC_CTS_INF {routing information }
6: $AODV\_CROSS = FALSE$
7: *END*

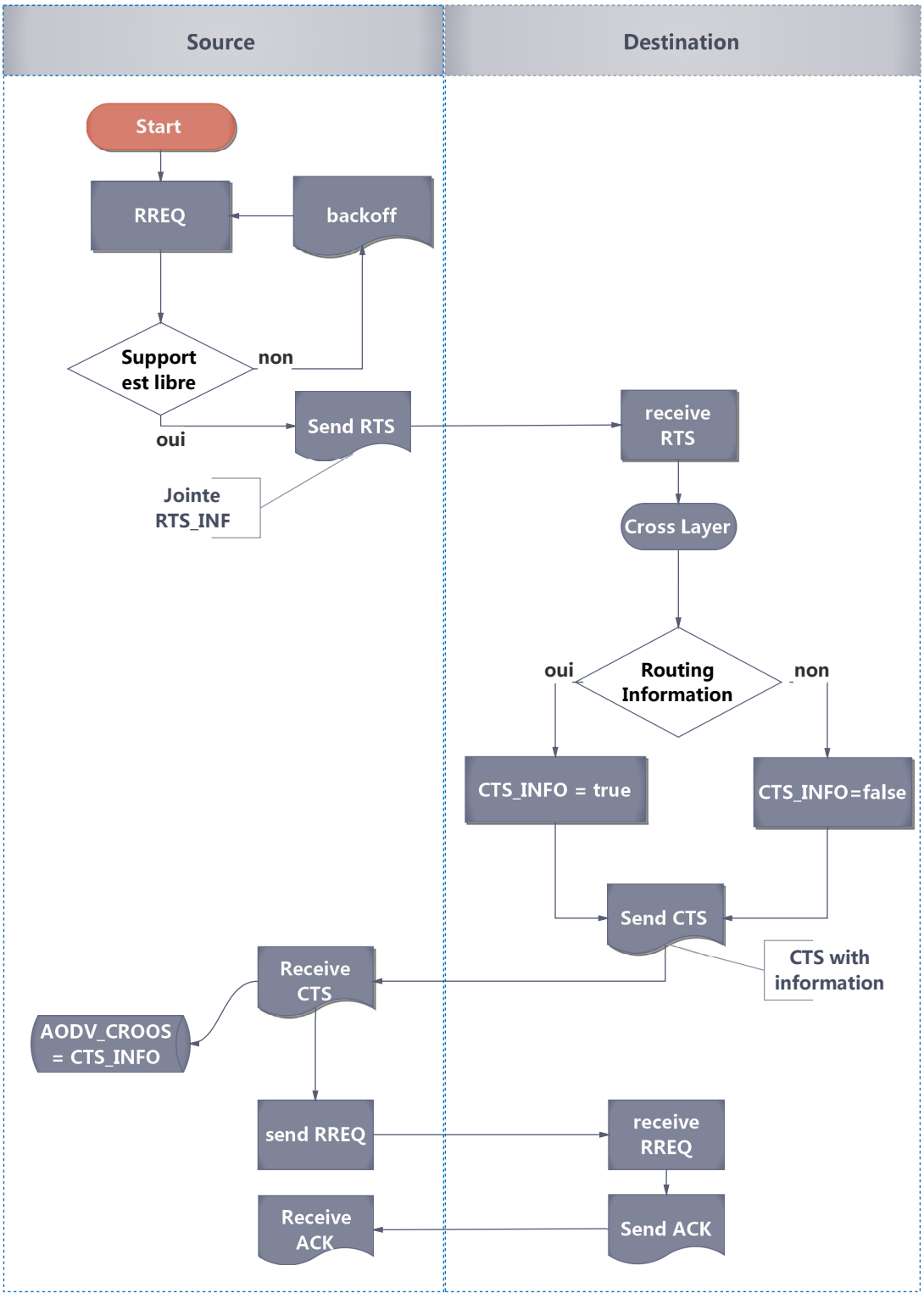

Figure 4.5 – extension de la découverte de route

4.4.2 Extension de la couche Réseau

Après la réception de paquet RREP le nœud source consulte les informations obtenues, le protocole AODV prend en considération dans sa décision de routage les deux facteurs(le numéro de séquence existant dans le paquet RREP et le nombre des sauts). Si la valeur de numéro de séquence est supérieure par rapport à la valeur de la requête, le nœud source doit mettre à jour sa table de routage.

Dans notre proposition, la validation du chemin se fait à travers une combinaison entre l'information obtenue avec le processus de vérification et les informations du paquet RREP. Par un accès inter-couche à l'information *AODV_CROSS*, le nœud source récupère l'information obtenue par le processus d'accès au support.

la décision finale de routage est prise suite à une combinaison avec les informations de routage récemment reçue sur le chemin recherché et l'information du *AODV_CROSS*.

Si l'information obtenue par le processus inter-couche est vraie, cela veut dire que le saut prochain possède des informations de routage sur la destination et permet de confirmer les réponses de routage. Dans le cas ou les deux informations (MAC et réseau) sont vraies le chemin est accepté, une misse à jour de table de routage est faite. Le nœud commence la transmission des données via la route ajoutée. Le détail est résumé dans la figure 4.6

Algorithm 4 Pseudo code of route reponse

```
AODV_CROSS {cross_layer information }
Source Receive RREP
if routing information == TRUE AND AODV_CROSS == TRUE
then
  Update routing table
  Start transmision
else
  if routing information == TRUE and AODV_CROSS == FALSE
  then
    DROP RREP /* Node source of RREP is a malicious */
    ADD RREP_Source_ID in malicious Table
  end if
end if
END
```

D'autre part, à la réception d'un paquet RREQ, le nœud *black hole* répond directement par un paquet RREP avec une valeur de numéro de séquence très grande, cela lui rend comme étant le saut qui amène vers la destination. L'attaquant *black hole* n'a pas une table de routage s'il reçoit un paquet RREQ va répondre directement par un paquet RREP avec un numéro de séquence grand, et ne joint aucune

information de vérification avec le processus de vérification *MAC_CTS_INF* qui est vide, dans ce cas l'émetteur détecte l'existence d'un nœud malveillant et élimine le paquet RREP. Dès que le nœud source détecte l'existence d'un nœud *black hole*, il lui ajoute à la table des nœuds malveillants.

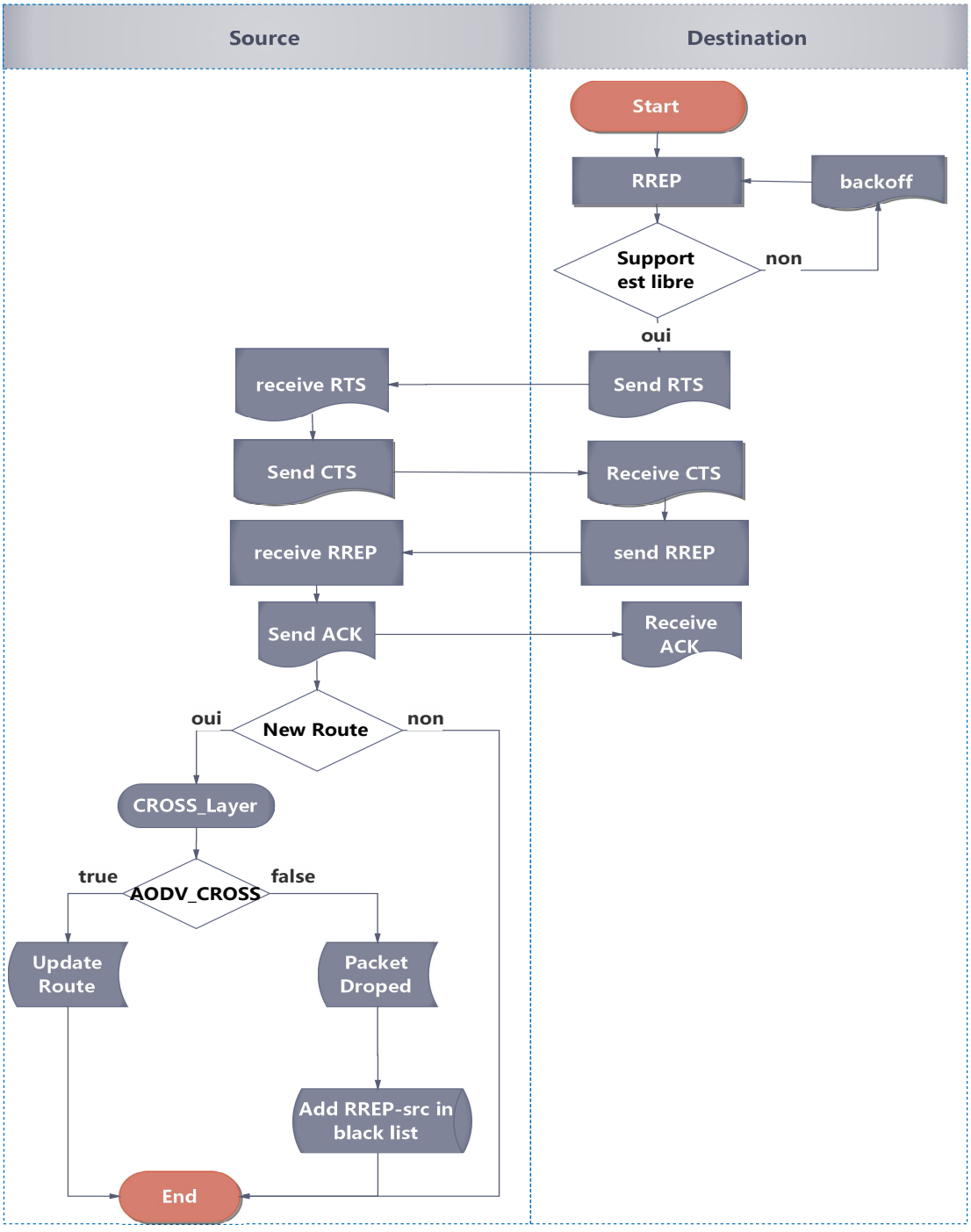

Figure 4.6 – extension de la réponse de route

4.5 Simulation

4.5.1 Environnement de Simulation

Un modèle de simulation a été développé en utilisant le simulateur de réseaux *ns2.34* où l'évaluation a été faite par une analyse des résultats sur les quatre protocoles ci-dessous :

1. protocole AODV normal
2. protocole AODV avec l'attaque *black hole*
3. les deux approches [Nital et al., 2010] et [Raj and Swadas, 2009]
4. notre proposition CrossAODV avec l'attaque *black hole*

Le point de cheminement aléatoire (RWP) [Bettstetter et al., 2003] est utilisé comme modèle de mobilité pour chaque nœud. Dans ce modèle, chaque nœud choisit une destination aléatoire à l'intérieur de la zone de simulation, ensuite, il se déplace à cette destination avec une vitesse aléatoire. Nous définissons les paramètres de simulation comme indiqués dans le tableau suivant :

Table 4.5 – Paramètre de simulation

Paramètres	Valeur
Nombre des nœuds	50
Nombre de nœud *black hole*	1..7
Le nombre de connexion	10,20,30
Type de trafic	CBR
la taille du paquet	512 Octets
Taux d'émission	4 paquets/s
Temps de pause	50 S
La vitesse maximal	5 MS
La dureé de simulation	200 S
La zone de simulation	500 x 500 M

4.5.2 Métrique d'analyse de performance

Dans notre approche nous avons utilisé les paramètres de mesure de performance les plus utilisés dans la littérature cités ci-dessus :

– Taux de délivrance de paquet (PDR) : Ce paramètre représente le pourcentage des paquets qui atteint la destination par apport aux paquets émis dans le réseau, et calculer comme suit :

$$PDR = 100 \times \frac{packetsreceive}{packetssend} en\%$$

– Délai de délivrance : C'est la moyenne de temps nécessaire pour délivrer un paquet à partir de la station source vers une station destination en incluant tous les retards dû au stockage dans le tampon *buffer*.
– Taux de Contrôle (Overhead) : le nombre de paquets de contrôle (RREQ, RREP, RERR) divisé par le nombre de paquets de données reçus . Ce critère permet de décrire le taux de contrôle nécessaire pour chaque paquet reçu .

4.6 Discussion Résultat

4.6.1 Taux de délivrance de paquets *Packet Delivery Ratio* (PDR)

La figure 4.7, présente l'évolution du taux de paquets délivrés avec succès dans les cas où les nœuds exécutent : AODV sans attaque, AODV avec attaque, et notre proposition crossAODV avec attaque comparé avec les deux approches [Nital et al., 2010] et [Raj and Swadas, 2009]. L'observation de cette figure montre l'évolution décroissante de PDR du protocole AODV sous l'attaque par rapport au protocole AODV normal, quand pause time = 0 la dégradation du PDR est de 32,96% Ceci est justifié par le fait que plus le pause time est minimal plus la topologie du réseau change de manière fréquente (les nœuds sont instables) et les nœuds malveillants ont moins d'opportunités d'intercepter les paquets de données qui ont atteint leurs destinations. Au fur et à mesure que le pause time croit les nœuds devient plus stables ce qui va diminuer légèrement le PDR jusqu'à 90,93%, à pause time = 200 par rapport au protocole AODV normal. Cette dégradation du PDR est prévisible dans la mesure où le nombre de paquets émis est largement supérieur au nombre de paquets reçus. Le nombre de paquets émis est important parce que tous les paquets de données reçus par les nœuds malveillants sont généralement ignorés. Nous constatons que notre système de détection et d'élimination crossAODV a amélioré le PDR par rapport au protocole AODV sous attaque.

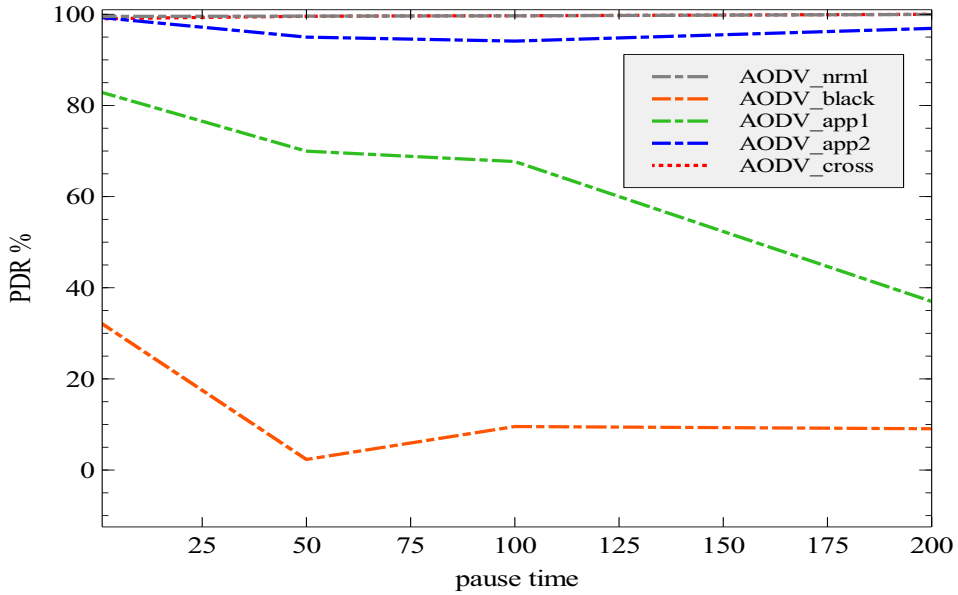

Figure 4.7 – taux de délivrance de paquet

4.6.2 Délai de délivrance de paquet

La figure 4.8 montre une évolution décroissante du délai moyen de bout en bout en fonction de temps de pause (pause time) Le déplacement des nœuds (pause time=0) implique des coupures fréquentes dans les chemins établis. Ils sont obligés de reconstruire les chemins invalides assez souvent ce qui conduit à l'obligation de découvrir un nouveau chemin c'est-à-dire tamponner les paquets et retardé la livraison de données ce qui signifie augmentation de délai. Au fur et à mesure que le réseau se stabilise (pause time = 200) le délai diminue. Ainsi, On constate que le délai requis pour CrossAODV est supérieur à celui d'AODV sous attaque. Effectivement crossAODV établit des routes sûre en évitant les nœuds malicieux cela à un impact sur le délai qui est plus que AODV normal en (pause time =200).

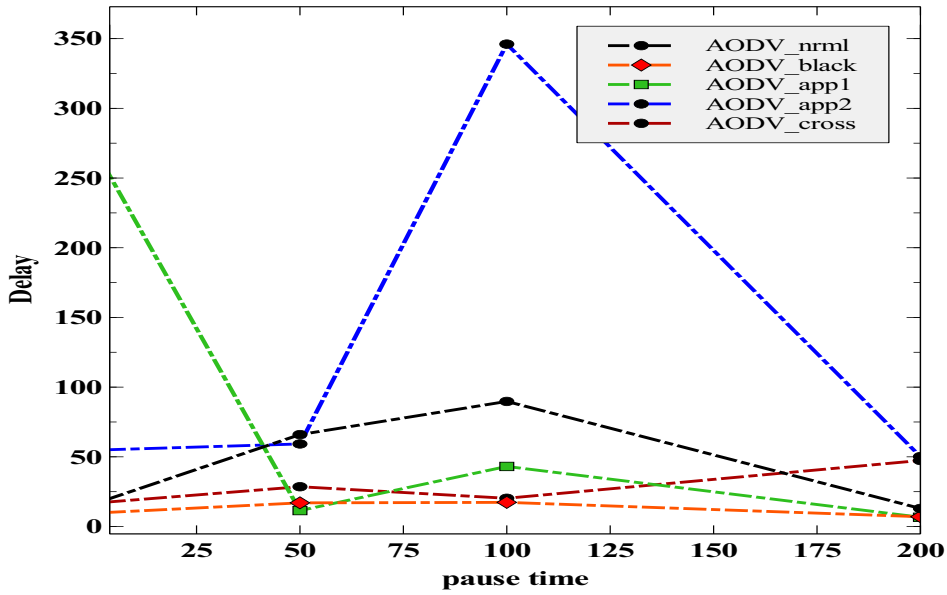

Figure 4.8 – délai de délivrance de paquet

4.6.3 Taux de Contrôle *Overhead*

L'observation de la figure 4.9 montre une évolution décroissante du trafic de contrôle en fonction de pause time. Nous remarquons que le protocole AODV sous attaque génère moins de trafic de contrôle que le protocole AODV classique. Ce phénomène s'explique du fait que les nœuds malicieux observent les RREQ et ne rediffusent pas car il y'a une forte probabilité de rupture des liens à pause time = 0 d'où la nécessité d'une nouvelle procédure de recherche de route. Au fur à mesure que le réseau se stabilise (pause time=200) les paquets de contrôle diminués.

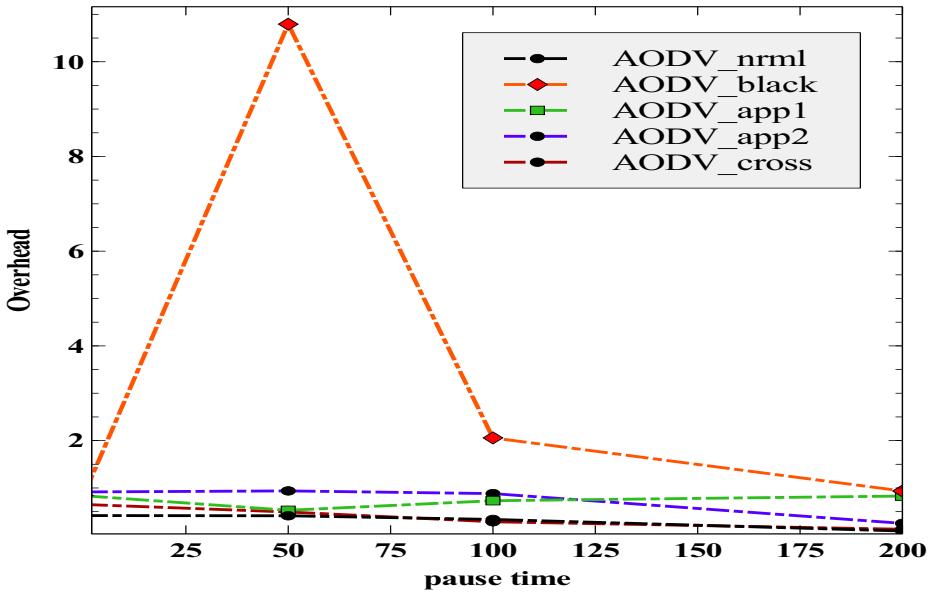

FIGURE 4.9 – taux de contrôle

4.6.4 Influence de nombre de connexion

La figure 4.10 montre que lorsque le nombre des connexions augment, le PDR diminue car il existe un grand nombre des connexions. Beaucoup de paquets de données sont perdus en raison de la surcharge et la saturation de la file d'attente de réseau dans AODV normal et crossAODV. Dans le protocole AODV sous attaque, le PDR diminue progressivement lorsque le nombre des connexions augmente, car un grand nombre des paquets sera ignoré par le nœud *black hole*.

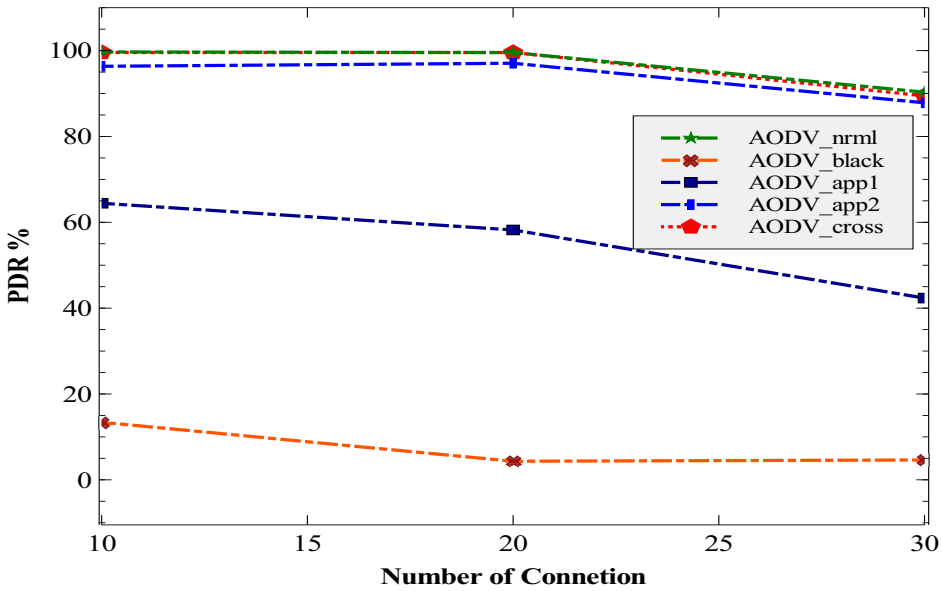

FIGURE 4.10 – Effet de nombre de connexion sur taux de délivrance de paquet

4.6.5 Influence de nombre nœuds *black hole*

La figure 4.11 montre que le PDR ne change pas quand le nombre de *black hole* augmente cela conclu que le nombre de nœud malveillant n'a pas d'incidence dans notre approche. La dégradation des PDR dans 30 connexions c'est à cause de la surcharge des réseaux et non pas à cause de nombre des nœuds *black hole*.

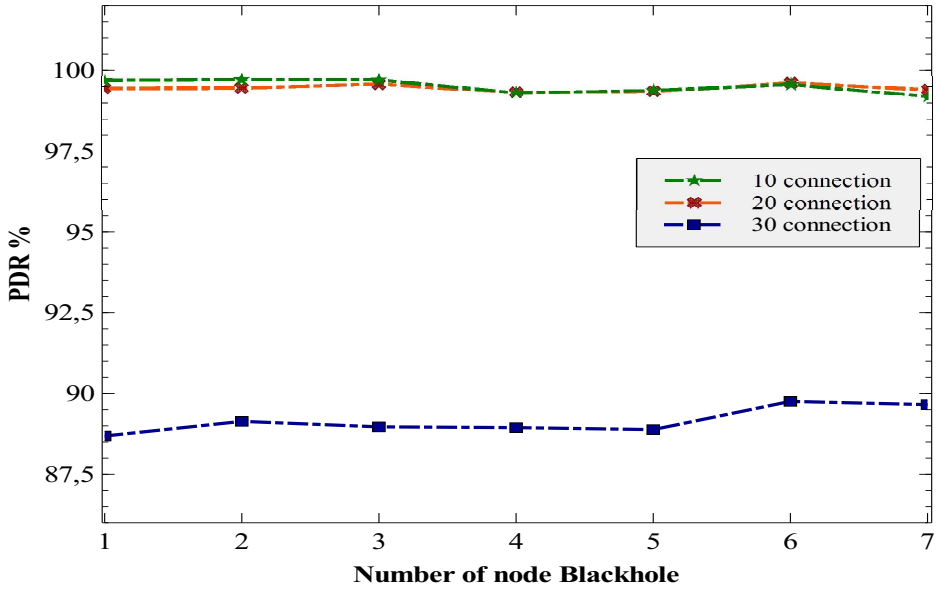

Figure 4.11 – Effet de nombre de *black hole* sur taux de délivrance de paquet

4.7 Conclusion

Dans ce chapitre, nous avons proposé une méthode appelée CrossAODV pour la détection et l'isolation des nœuds malveillants qui utilisent l'attaque de trou noir *black hole* dans le protocole AODV. Cette méthode est basée sur la coopération entre la couche réseau et la couche d'accès au support en exploitant la fonction de coordination distribuée *DCF*. L'approche se compose de deux processus : la vérification et la validation. Lors de la découverte de route, le processus de vérification utilise les deux paquets RTS / CTS pour récupérer des informations sur le chemin demandé. Le processus de validation consiste à demander la même information et en comparant les informations de routage demandées avec l'information obtenue durant la phase de vérification.

La méthode proposée a été analysée et comparée avec des travaux connexes à l'aide de différents paramètres de performance tels que le taux de délivrance de paquets, le délai de bout en bout, et le taux de contrôle. Comme illustré dans les résultats, nous pouvons facilement conclure que la performance de notre approche est meilleure par rapport aux travaux connexes.

Notre solution CrossAODV assure une augmentation de PDR avec une petite addition négligeable dans le taux de contrôle. Tandis que, le nombre de connexion à une influence sur le PDR, puisque il y'a assez de paquets de données qui sont transmises, cela conduit à une quantité de paquet perdu suite au surcharge de la file d'attente. la variation dans le nombre des nœuds malveillants ne modifie pas notre approche.

Chapitre 5

Système amélioré à base de réputation pour détecter les nœuds malveillants dans MANETs

Contents

5.1 Introduction

La sécurité des réseaux Ad Hoc présente un défi global. En effet les MANETs possèdent des caractéristiques qui les rendent plus vulnérables aux attaques passives et actives. Il existe plusieurs attaques qui visent les réseaux ad hoc pour perturber ou dégrader leurs performances. Un nœud présentant un comportement malveillant supprime les paquets de données qu'il reçoit sans les transmettre vers leur destination désirée. À partir de ce point, il est nécessaire de concevoir des solutions pour lutter contre ces attaques, parmi beaucoup de solution on a ce qu'on appelle les systèmes basés sur la réputation. Cependant, dans un système de réputation, le nœud surveille le comportement de ses voisins sur toutes les communications émises et reçues. Ensuite il calcule une valeur de réputation. La valeur peut être calculée directement *(first hand experience)* ou via une combinaison de réputation directe et indirecte *(second hand experience)*. Dans ce chapitre nous présenterons notre deuxième proposition basée sur la réputation directe, Nous commencerons d'abord par présenter les motivations de cette proposition (les techniques basées sur l'écoute et les techniques basées sur l'acquittement)[Senthilkumar and William, 2014]. Ensuite nous présenterons notre amélioration qui consiste à comptabiliser les paquets de données non transmis à cause des circonstances comme l'épuisement d'énergie, la surcharge de la file d'attente et la mobilité du nœud voisin. Enfin, nous analyserons ses performances.

5.2 Les systèmes de réputation

Des nombreux mécanismes intéressants ont été étudiés et mis au point pour traiter différents systèmes basés sur la réputation dans les réseaux ad hoc. L'idée principale du système de réputation est que chaque nœud surveille l'activité de ses voisins (émetteur et le mode récepteur). Ce système est composé de trois phases principales : la surveillance, la gestion de la réputation et de l'isolement. Un nœud normal coopère dans les processus de communication pour que sa réputation a une valeur maximale contre le nœud malicieux qui a une valeur faible [Akhtar and Sahoo, 2013]

La valeur de réputation calculée est comparée avec une valeur de seuil pour déterminer un nœud malveillant, elle est mise à jour dans chaque intervalle de temps [Wang et al., 2010]. Il existe un système de gestion de la réputation qui utilise à la fois la première main et des informations de seconde main pour mettre à jour les valeurs de réputation. Le premier obtenu directement par la phase de surveillance, et le deuxième désamorcée par le nœud voisin [Han et al., 2014]. Certaines des

approches efficaces vont être énumérées par la suite.

5.2.1 Approches à base d'écoute

Les auteurs de [Bansal and Baker, 2003] définissent une approche à base d'un système de réputation dans les réseaux ad hoc nommés OCEAN *(Observation-based Cooperation Enforcement in Ad Hoc Networks)*. Leur méthode se focalise sur l'observation directe des transmissions des nœuds voisins. Chaque nœud estime et maintient une valeur de réputation sur ces voisins, il déploya un routage basé sur cette valeur. Chaque nœud tamponne une copie du paquet transmis et surveille le comportement de ses voisins, Si le voisin qui a reçu le paquet ne le retransmet pas, à ce moment sa valeur de réputation est diminue. Chaque nœud apparait dans la liste noire du nœud malveillant est évité dans la découverte de route. OCEAN résout le problème de la fausse réputation échangée par les voisins (second hand), elle donne une deuxième chance pour les nœuds malveillants de prouver leurs fiabilités et changer leurs mauvais comportements.

CONFIDANT : (cooperation of nodes : fairness in dynamic ad hoc networks) a été proposé par [Buchegger and Le Boudec, 2002].

C'est un système hybride basé sur la confiance, qui a été développé dans le protocole de routage DSR pour détecter et supprimer les nœuds malveillants. Il est composé de quatre éléments principaux le moniteur, le gestionnaire de réputation, le gestionnaire de chemin d'accès et le gestionnaire de confiance. Ce système utilise la méthode bayésienne pour le modèle adaptatif. Il combine la réputation de la première phase et de la seconde phase pour gérer une valeur finale. Dans CONFIANT, le gestionnaire de chemin décide le chemin qui contient les nœuds avec une faible réputation sinon le gestionnaire de confiance maintient les réputations obtenues par les voisins. Le gestionnaire de réputation donne un poids élevé pour une réputation calculer par la première phase et un faible poids pour la réputation obtenue par la deuxième phase. Dans ce système, il n'y a aucun moyen de garantir l'intégrité de la réputation échangée entre les nœuds. Si un nœud malveillant est détecté, un message d'alarme contient l'identité de ce dernier est envoyé vers tous les nœuds voisins pour les avertir sur ce nœud.

[Marti et al., 2000] ont proposés une technique basée sur la réputation qui emploie un système de surveillance et un évaluateur de chemin. Le système de surveillance écoute les paquets transmis par le saut suivant, Si une correspondance est trouvée, le paquet est retiré de la mémoire tampon, et le nœud est déterminé comme étant un nœud normal. Dans le cas où le saut suivant ne transmit pas les paquets reçus, ou les paquets tamponnés dépassant une période de temps, ce dernier est compté

comme étant malveillant. Si un nœud malveillant est détecté le nœud de source doit être notifié ainsi que les nœuds voisins. L'évaluateur de chemin choisi le chemin qui ne possède pas des nœuds malveillants pour transmettre les données.

Un mécanisme de réputation et de collaboration pour respecter la coopération du nœud dans les réseaux mobiles ad hoc a été proposé par Michiardi et Molva [Michiardi and Molva, 2002]. CORE emploie trois types de réputation (subjective, indirecte et fonctionnelle). La réputation subjective est obtenue par la première phase par contre la réputation indirecte est obtenue par les voisins. La réputation fonctionnelle est calculée à partir de la réputation précédente en utilisant des poids différents, où chaque poids correspond à chaque critère d'évaluation (exemple l'envoi de données a son poids). La réputation finale est estimée en combinant les différentes réputations fonctionnelles. CORE permet aux nœuds d'échanger seulement les réputations positives. CORE est une méthode vulnérable aux fausses réputations distribuées.

[Gong et al., 2010] ont proposé un modèle qui détecte avec succès des nœuds malveillants et les éviter dans le chemin de routage. Les nœuds contrôlent les activités de ses voisins avec l'écoute du trafic de transmission, Si le paquet est transmis par le voisin, donc son vecteur de confiance local est évalué. Chaque nœud vérifie l'intégrité des paquets d'une façon que le paquet n'a pas été modifié. Si l'intégrité de paquet n'a pas été altéré et le paquet a été bien transmis, alors un vecteur de confiance est calculé. Chaque nœud dans ce modèle fonctionne comme étant observateur et contrôleur des paquets de données qui sont transmises dans sa zone de couverture, ainsi il note les transmissions réussies par ses nœuds voisins. Si un nœud voisin ne contribue pas dans un intervalle de temps, ce nœud sera testé pour son comportement suspect.

Les auteurs de [Ayday and Fekri, 2012] proposent un système utilisé dans des environnements pair à pair (P2P). Leur technique permet de distribuer un message pour évaluer la réputation et la fiabilité d'un nœud dans le réseau. La réputation des nœuds est basée sur la qualité de service dans la réception des paquets fournis par le serveur, et la fiabilité est notée après chaque transmission réussie.

Patel et al dans [Patel and Jhaveri, 2016] ont proposé une méthode qui combine la phase de découverte de route et la phase de transmission de données pour détecter le nœud malveillant. Dans l'itinéraire pendent la phase de découverte, si le numéro de séquence de destination dépasse une valeur de seuil, alors le paquet RREP est rejeté. Plus tard au cours de la phase de transmission de données, le nœud calcule la différence entre les paquets transmis et reçus. Si cette différence dépasse une valeur de seuil, alors le nœud est considéré comme malveillant. L'identité de nœud

malveillant est distribuée aux nœuds voisins et une liste noire est maintenue.

5.3 Notre approche

5.3.1 Modèle de réseau

Dans notre contribution nous avons modélisé le réseau MANET comme un graphe dirigé $DG(N, E)$, où $N = N_1, N_2, ..., N_k$ représente l'ensemble de K nœuds mobiles et E représente les liaisons de communication entre chaque paire des nœuds dans la même zone de transmission. La Valeur de réputation du nœud N_j calculée par le nœud N_i sera noté par R_{ij}. Nous supposons que les communications sont bidirectionnelles dans chaque lien de communication, cela signifie que chaque nœud peut émettre et recevoir même aussi écouter ce qu'il circule dans sa zone de couverture. Toutes les notations utilisées dans cette proposition sont résumées dans le tableau ci-dessous 5.4.1.

Table 5.1 – Les notations utilisées

Notation	Description
NGS_i	Ensembles des nœuds voisins de N_j
N	Nœuds $N_s, .., N_i, N_j, ..N_d$ sachant que N_s est la source N_d est la destination
PNT_j	Le nombre de paquet transmis par N_i et non retransmis par N_j
PSN_j	Le nombre de paquet transféré par N_j
PER_j	Le nombre de paquet non transmis a causse d'erreur N_j
$RECV_j$	Le nombre de paquet reçu par N_j
R_{ij}	Réputation de N_j maintenue par N_i
R_t	Temps de calcul du réputation
R_{exp}	Temps d'expiration du réputation
R_{init}	La réputation initiale

5.3.2 Modèle de nœud malveillant

Dans notre proposition, nous définissons un nœud de comportement malveillant, chaque nœud qui coopère dans le processus de découverte de route avec un chemin parfait vers la destination. Dès que ce nœud introduit dans la liaison de transmission des données actives, il commence d'agir avec les actions suivantes [Dini and Duca, 2012; Akhtar and Sahoo, 2013] :

1. Suppression des paquets RREQ : le nœud malveillant refuse de relayer les paquets RREQ dans le processus de découverte de route et d'éviter de coopérer

dans ce processus.

2. Pour chaque RREQ reçu le nœud malveillant répond par un paquet RREP qui contient une nouvelle information optimale comme le numéro de séquence et le nombre de saut.
3. Le nœud malveillant supprime tous les paquets de données qui les traversent et perturbe le processus de transfert de données.

5.3.3 Le système proposé

Lorsque nous avons comparé certains travaux connexe à notre modèle proposé, une différence importante est dans la technologie de surveillance qui est utilisée pour la détection des nœuds malicieux.

La majorité des techniques utilisent un nœud de surveillance *(watchdog)* qui entend les paquets transmis, reçus et non retransmis pour calculer les valeurs correspondantes de la réputation. Cette méthode a ses propres limites. Dans les MANETs, la transmission est sujette à de nombreuses circonstances comme la collision des paquets, la surcharge de la file d'attente et l'inaccessibilité de la destination à cause du mouvement ou d'épuisement d'énergie. Tout cela peut mener à une détection erronée ou de mauvaises valeurs de réputation.

Dans notre travail nous avons proposé un système de réputation (voir figure 5.1), et nous avons amélioré la phase de surveillance, il n'est pas basé uniquement sur la transmission de données (envoyer et recevoir) pour calculer la réputation de ses voisins. Mais toutes les circonstances du nœud seront prises en compte telles que la surcharge de la file d'attente, rupture de lien et épuisement d'énergie. L'objectif de notre proposition est d'identifie et d'isoler le nœud malveillant, Par suite le processus se compose de trois phases :

1. La surveillance
2. Calcul de la réputation
3. Isolation et maintenance de route

La Surveillance ou L'écoute

C'est la phase responsable de l'observation directe de chaque nœud voisin N_j sachant que N_j appartient à NGS_i. Chaque nœud entend les paquets des données transmis et reçus par ses voisins. Nous adaptons à notre système de surveillance une amélioration, où le nœud attend pendant un intervalle de temps pour savoir si le paquet est transmis au prochain saut. S'il est transmis, la réputation maintenue pour

le nœud est augmentée ; Sinon, elle diminue. La valeur de réputation est recalculée périodiquement, en utilisant l'équation (5.3).

La collaboration du nœud voisin CL_j est calculée par le rapport de nombre des paquets des données transmis sur le nombre reçus. Le nœud malveillant supprime tous les paquets reçus et ne les transmis pas vers leurs nœuds voisins. Mais pratiquement il existe d'autres conditions qui provoque la suppression du paquets tels que la surcharge de la file d'attente ou la défaillance du chemin qui mène vers le saut suivant.

Dans notre méthode, nous avons ajouté une amélioration pour assurer la cohérence de la congestion et inaccessibilité du saut suivant, si le nombre de retransmission atteint le maximum *MAC_RETRY_COUNT_EXCEEDED*, le paquet sera supprimé en raison de la surcharge de la file d'attente par la couche mac. Le nœud indique à la couche supérieure qu'il y a un échec d'opération via la variable *xmit_failure_data_*. Dans ce cas, nous avons ajouté un compteur d'erreur (PER) qui influence à la réputation de nœud. Pour distinguer entre la surcharge de la file d'attente et l'inaccessibilité du nœud, nous avons ajouté un champ dans le paquet *CTS* pour avoir deux informations (l'énergie résiduelle et l'état du surcharge de la file d'attente).

Système de réputation :

Dans notre approche, nous calculons une valeur de réputation R_{ij} par nœud N_i pour le nœud N_j concernant leur données transmises. Nous avons utilisé pour détecter les nœuds malveillants un ensemble des compteurs pour chaque nœud voisin N_j de l'ensemble NGS_i. Lorsque le nombre des paquets reçus en N_j atteint des valeurs de seuil P_j, dans ce cas, on peut juger leur comportement.

La valeur de réputation R_{ij} est un nombre réel, Une valeur positive indique que le nœud est coopératif alors qu'une valeur négative ou nul indique que le nœud se comporte mal. La réputation d'un nœud est maintenue par ses voisins jusqu'à une période de temps R_{exp}. L'ensemble des compteurs est mis à jour comme suit :

- Règle 1 : Pour chaque paquet reçu par N_j et relayé, le PSN_j est incrémenté par un.
- Règle 2 : Pour chaque paquet reçu par N_j et non relayé, le PNT_j est incrémenté par un.
- Règle 3 : Pour chaque paquet reçu par N_j et non relayé en raison d'une surcharge de file d'attente ou de coupure du lien, le PER_j est incrémenté de un.

$$R_{ij} = CL_j - PNF \tag{5.1}$$

$$R_{ij} = \left(\frac{PSN_j}{RECV_j} + \frac{PER_j}{RECV_j}\right) - \left(\frac{PNT_j}{RECV_j}\right) \tag{5.2}$$

Étant donné $PNT_jPSN_jPER_j$ nous calculons la réputation du nœud voisin N_j périodiquement R_t.

PNF représente le rapport des paquets non relayés. Il reflète la mauvaise coopération du nœud. Après le temps d'expiration, chaque nœud met à jour la réputation de ses voisins selon l'équation (5.3). Le nœud qui rejoint le réseau reçoit une réputation neutre R_{init}.

$$R_{ij}^t = \alpha \times R_{ij}^{t-1} + (1 - \alpha) \times R_{ij}^t \tag{5.3}$$

Nous avons défini $\alpha = 0.4$ pour donner plus de chance au nœud malveillant pour améliorer sa réputation et le seuil de paquet avant de juger un nœud qui a été fixé à $P_j = 3$ paquets ses valeurs ont été choisies suite à des tests d'expérimentations. L'algorithme 5 résume la phase de calcul de réputation :

Algorithm 5 Calcul de réputation

```
1: BEGIN
2: FOR each data packets send by N_i and received at N_j do
3: if Nj == N_d then
4:    Send DATA
5: else
6:    Send DATA
7:    Update RECV_ij
8: end if
9: if RECV_ij < P_j then
10:   if Nj transmit DATA then
11:      Update PSN_j according rule (1)
12:   else
13:      if Nj send RERR to N_i then
14:         Update PER_j according rule (3)
15:      else
16:         UPDATE PNT_j according rule (2)
17:         Packets is dropped
18:      end if
19:   end if
20: else
21:   Compute R_ij according equation (1)
22:   if R_ij < 0 then
23:      Node is malicious
24:      Call Isolation phase
25:   else
26:      Continue The transmission
27:   end if
28: end if
29: END
30: END
```

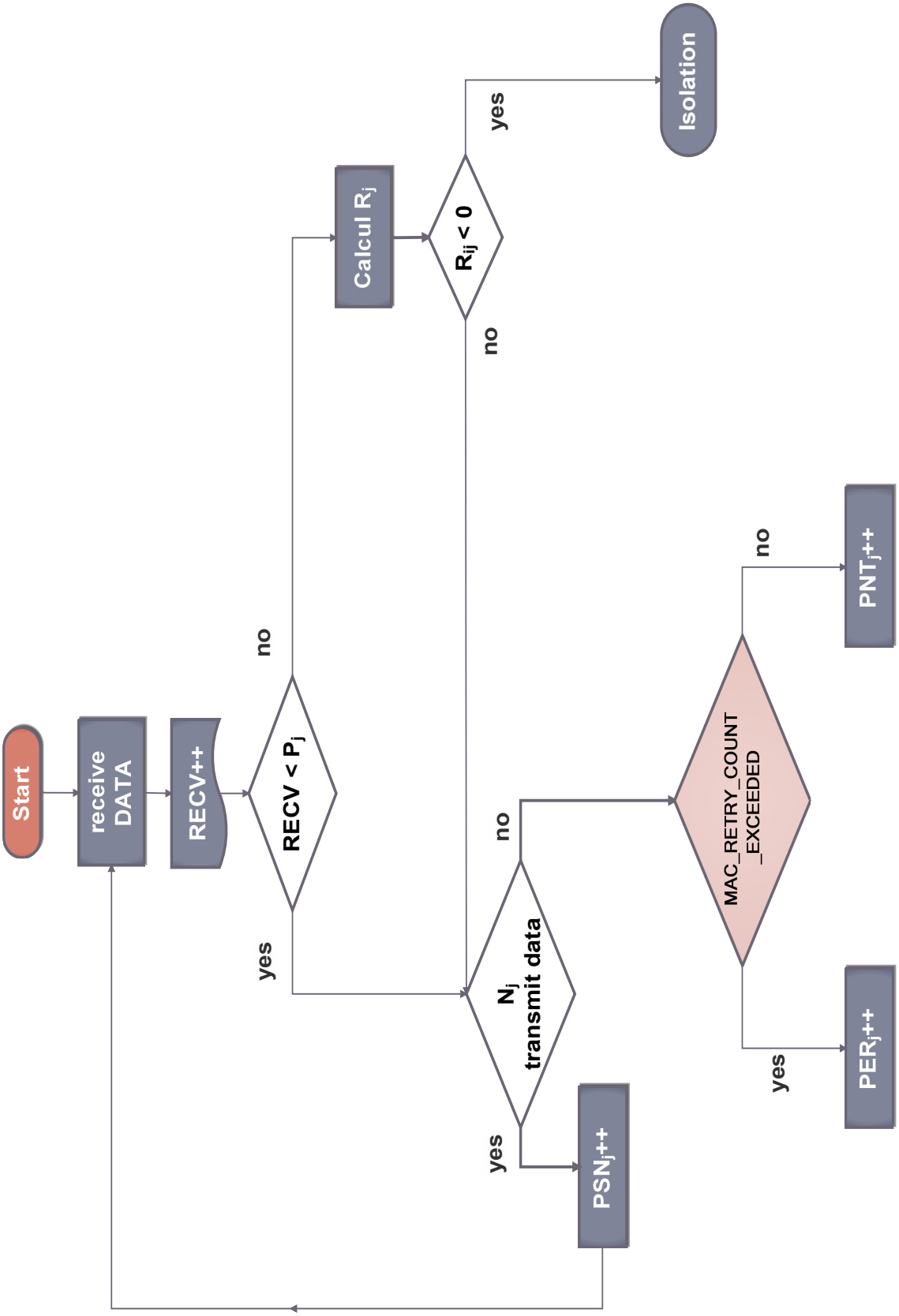

Figure 5.1 – Système de réputation amélioré

Isolation et maintenance de route :

Le nœud avec une réputation inférieure à zéro est considéré comme malveillant, un processus d'isolement sera lancé 5.2. Le nœud de surveillance arrête la transmission de tous les paquets passant par ce voisin qui a un mauvais comportement. Ensuite, il ajoute ce nœud malveillant dans une liste noire *Black_list* pour un certain temps de validité T_{exp}. Si ce nœud malveillant a été détecté d'autre fois à cause de son comportement, le compteur TH sera incrémenté jusqu'à atteindre ou dépasser une valeur de seuil $Th = 3$, cela signifie que le même nœud est détecté plusieurs fois, dans ce cas, le nœud ne sera jamais supprimé de la liste noire pour une période de temps infinie T_{infiny}. Ce compteur Th donne d'autre chance au nœud malveillant pour changer son mauvais comportement. Le nœud détecteur informe tous les voisins avec un paquet de contrôle "*Alert*" vers tous les voisins pour éviter ce nœud malveillant. En évitant le nœud malveillant une réparation locale est démarrée sinon une réparation globale.

Dans la maintenance de route nous avons amélioré la réparation locale de chemin avec deux informations (l'énergie résiduelle et l'état de surcharge de la file d'attente). Nous choisissons le prochain saut avec une énergie résiduelle maximale et une faible surcharge de la file d'attente. Le pseudo code 6 illustre le fonctionnement de cette phase :

Algorithm 6 Isolation

```
1: BEGIN
2: if N_j in Black_list then
3:    if Counter > Th then
4:       Set T_exp = infinity
5:       Send Alert to neighbors
6:    else
7:       Set Texp = time slot
8:    end if
9: else
10:   Add N_j to Black_list
11:   Set T_exp = time slot
12:   Counter++
13: end if
14: Stop data transmission
15: Buffered DATA
16: Route repair
17: END
```

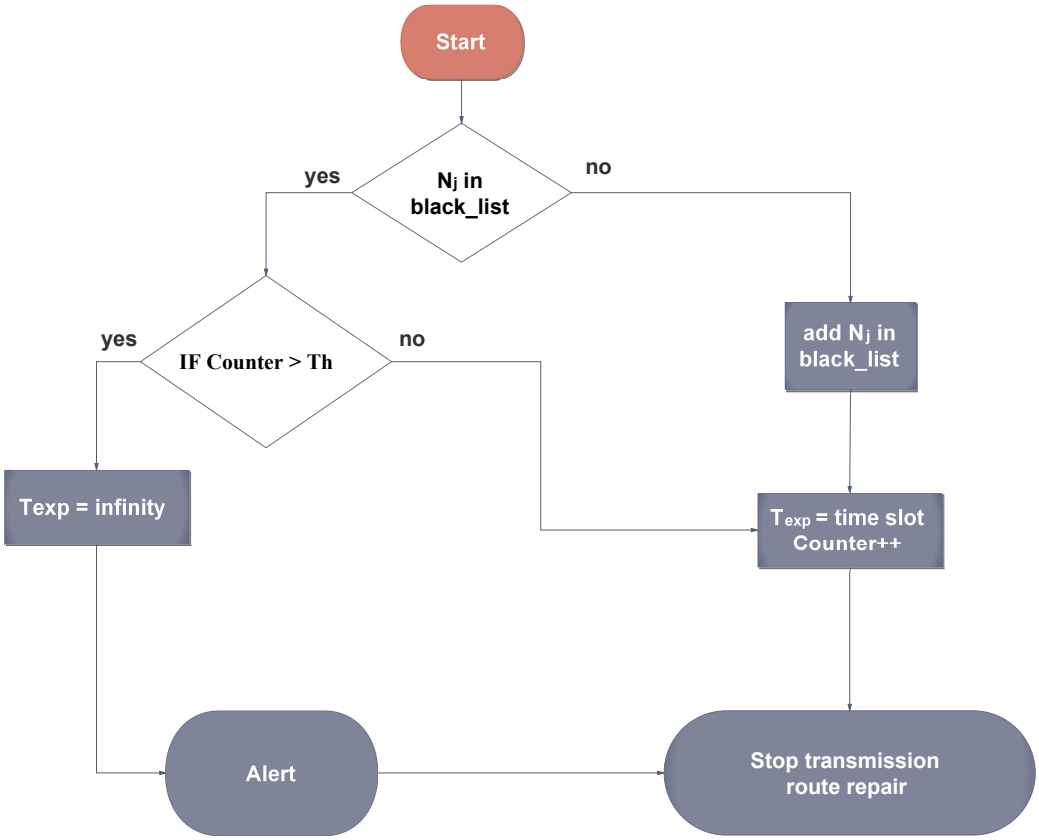

Figure 5.2 – organigramme de la phase d'isolation

5.4 Évaluation et Discussion

Dans cette section, nous effectuons une étude de simulation pour montrer l'intérêt de notre proposition et son efficacité.

5.4.1 Environnement de simulation :

Nous avons utilisé Network Simulator (NS2.35) pour simuler la méthode proposée. Nous utilisons la fonction de coordination distribuée (DCF) de l'IEEE 802.11 pour les réseaux locaux sans fil en tant que protocole de couche MAC pour notifier la couche réseau sur la défaillance de la liaison. Dans notre simulation, les nœuds mobiles se déplacent dans une zone de $500 \times 500m$ pour un temps de simulation de 200 s avec la même zone de transmission de 250 m. La vitesse maximale des nœuds est de 10 m/s. Nous avons varié le nombre de nœuds jusqu'à 50 et le trafic simulé est le débit binaire constant (CBR). Nous supposons que chaque nœud se déplace indé-

pendamment avec la même vitesse moyenne avec le modèle RWP [Bettstetter et al., 2003]. Nos paramètres de simulation sont résumés dans le tableau 5.4.1. L'évaluation a été effectuée en analysant les résultats de trois protocoles ci-dessous,

- Utilisation du protocole normal AODV
- Utilisation du protocole AODV avec un mauvais comportement de nœud
- Utilisation de notre approche avec un mauvais comportement de nœud

Table 5.2 – Paramètres de simulation

Paramètre	Valeur
Nombre des nœuds	51
Type de trafic	CBR
la taille du paquet	512 Octets
Taux d'émission	4 paquets/s
Couche MAC	802.11 S
La dureé de simulation	200 S
La zone de simulation	500 x 500 M

5.4.2 Les métriques de performance

Pour les métriques de performance, nous avons testé cette contribution selon les mêmes métriques utilisées précédemment dans la section 4.5.2

5.4.3 Discussion sur les résultats

La Figure 5.3 représente l'évolution du Ratio de Livraison de Paquet (PDR) avec le nœud de mauvais comportement, sans nœud de mauvais comportement et notre proposition avec le nœud malveillant en fixant le nombre de sources à 10, 20 et 30 sources de connexions. Nous remarquons que la méthode proposée produit des gains de performance significatif par rapport au protocole sous un comportement malveillant, ce qui justifie que la détection et l'isolement basés sur la réputation ont fonctionné parfaitement. Dans le temps de pause = 0 la dégradation de la PDR est de 20,21% dans notre proposition, Ceci est justifié à cause de nombre des paquets supprimés par le nœud de mauvais comportement en raison de la technique utilisée par notre système, car notre technique utilise un nombre de paquets qui vont être reçus par le nœud malicieux avant de juger sa réputation. Une autre raison, quand le temps de pause est petit la topologie de réseau change fréquemment (les nœuds sont instables), ce qui augmente le nombre des paquets perdu. Quand les nœuds sont très stables, cela réduira légèrement le PDR à 3,65% dans notre approche par rapport au

protocole normal. Nous observons aussi que le PDR sous attaque diminue de 90,91% par rapport au protocole sans attaque.

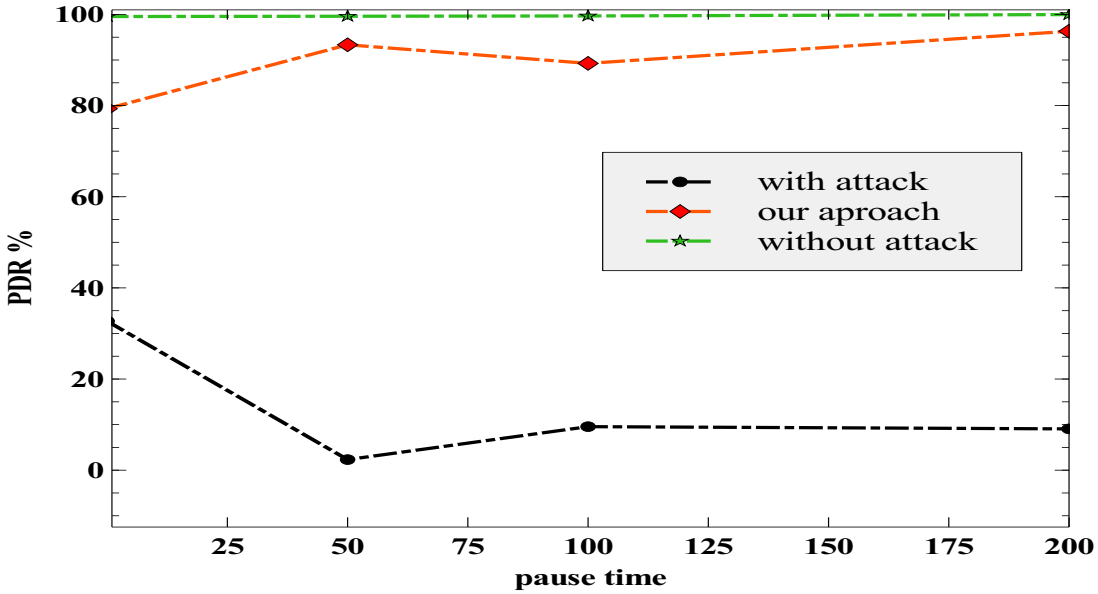

Figure 5.3 – taux de délivrance des paquets

L'observation de la figure 5.4 montre une augmentation dans le trafic de contrôle basée sur le temps de pause. Nous notons que le protocole AODV sous attaque génère moins de trafic de contrôle par rapport au protocole standard AODV, Ceci à cause du nœud malveillant qui reçoit les paquets RREQ et ne les rediffuse pas.Aussi il y a une forte rupture des liens dans Pause time = 0, ce qui provoque l'appel de la procédure de recherche de chemin. Lorsque le réseau se stabilise (pause = 200) le taux des paquets de contrôle diminue. Notre approche produit un trafic de contrôle légèrement élevé contre le protocole normal, car notre méthode génère d'autres paquets de contrôle d'alerte, ce paquet utilisé pour informer les voisins sur le nœud de mauvais comportement. Outre, cette augmentation du paquet de contrôle est dû au processus de réparation local.

La figure 5.5 montre le résultat du *Average End to end delay* en fonction d'un temps de pause. Nous observons que le *Average End to end delay* est affecté par notre système basé sur la réputation à (pause = 0) car les paquets de données sont mis dans la mémoire tampon lors de la détection du nœud de mauvais comportement jusqu'à ce que la réparation locale se réalise. D'ailleurs le délai de notre proposition surpasse ceux de AODV et AODV sous un comportement malveillant, car les nœuds sont obligés de reconstruire les chemins invalides, par conséquent, le nœud est obligé de trouver un nouveau chemin. Lorsque le réseau se stabilise (pause time = 200), le délai est diminué, pour une raison que notre approche a établi un itinéraire fourni

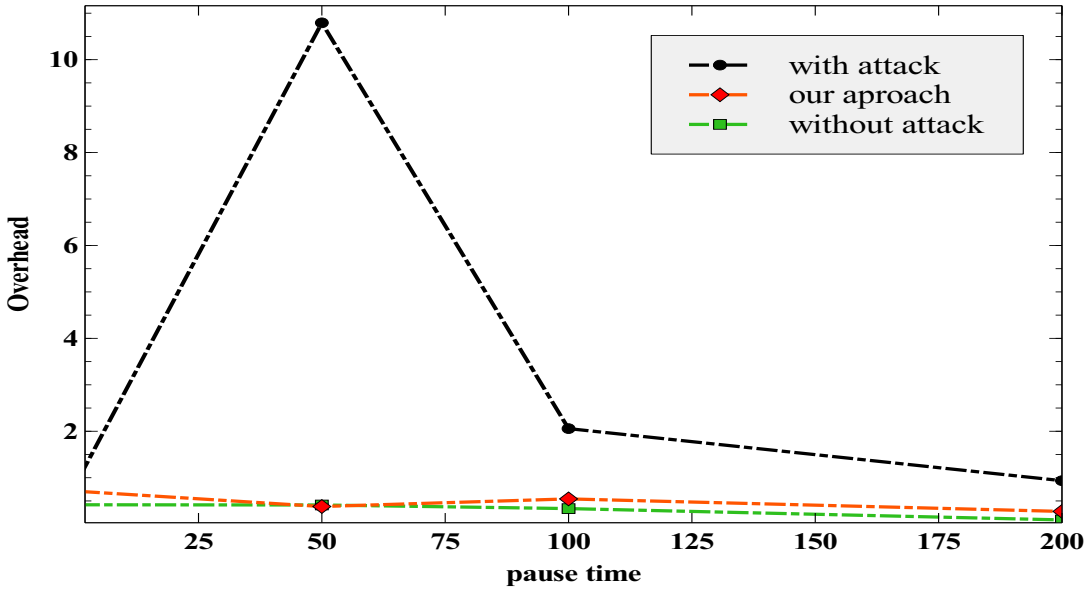

FIGURE 5.4 – taux de contrôle

pour éviter un nœud malveillant par la technique basée sur la réputation.

L'effet de α et P_j dans le taux de réussite du nœud de mauvais comportement :

Dans cette expérience, le taux de réussite du malveillant a été évalué en variant la valeur de α. Nous observons dans la figure 5.6 (a) que, si la valeur α augmente, le taux de réussite de nœud avec un comportement mauvais augmente lorsque $\alpha \geq 0,4$. En outre, lorsque $\alpha = 0$, le taux de succès d'un nœud malveillant est plus petit, cela signifie que la valeur de réputation du nœud est basée uniquement sur la valeur récemment calculée conformément à la règle (5.3). Si la valeur de $\alpha = 1$, dans ce cas, la valeur de réputation est négative et le taux de réussite est plus élevé sachant que le nœud à changer sa réputation. Dans la figure 5.6 (b), nous observons que si la valeur de P_j augmente le taux de réussite d'un nœud de mauvais comportement augmente parce que P_j représente le nombre des paquets reçus avant de juger le comportement du nœud en générale.

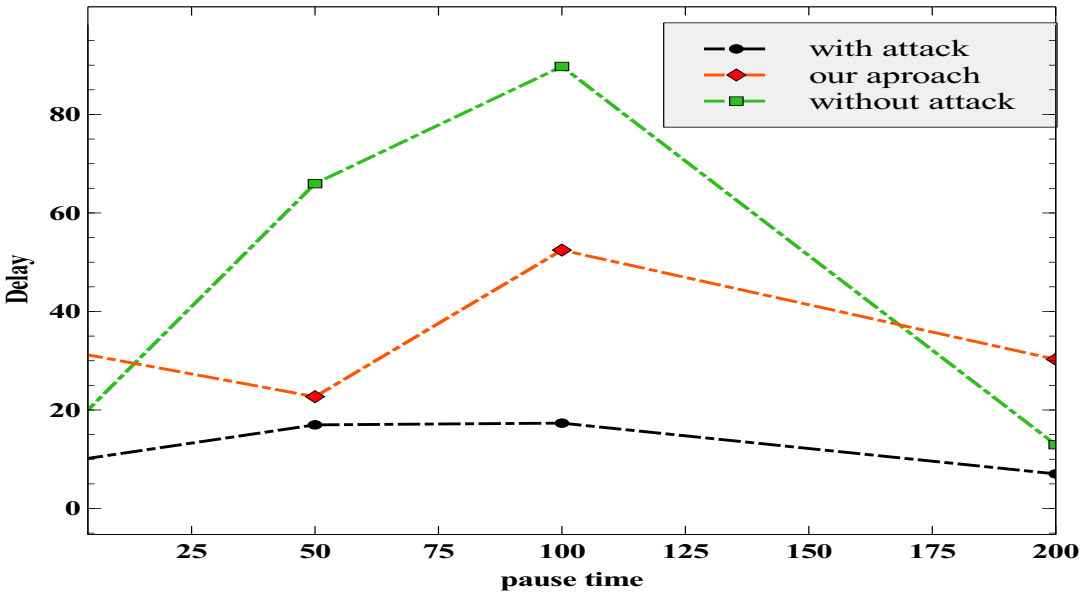

Figure 5.5 – délai de bout en bout

5.5 Conclusion

Dans ce chapitre, nous avons présenté une méthode de réputation améliorée pour détecter et isoler le nœud de mauvais comportement dans les réseaux mobiles ad hoc. Notre approche est composée de trois phases. Tout d'abord, la phase de surveillance est définie par l'écoute directe des paquets envoyés et reçus (first hand experience). Si le nombre de paquets reçus par le saut suivant est égal à une valeur de seuil, on peut à ce niveau juger ce nœud.

Deuxièmement, les calculs de réputation sont améliorés par un paquet d'erreur généré en raison de la surcharge de la file d'attente et l'indisponibilité des sauts suivants, cette information peut être distincte dans les paquets supprimés avec le nœud de mauvais comportement et par d'autres événements. Le nœud qui possède une réputation négative sera isolé, ainsi, les routes qui incluent ce nœud sont réparées. La réparation d'itinéraire prend en compte d'autre paramètre de *QoS* tel que le degré de surcharge de la file d'attente et l'énergie résiduelle pour établir un nouveau chemin.

Les résultats des simulations montrent que notre système de réputation surpasse AODV sous un nœud malveillant. Par conséquent, une augmentation négligeable dans le taux de contrôle est obtenu (presque 2 %) ; Également une augmentation de taux de livraison des paquets par rapport le protocole avec un nœud malveillant. aussi, un gain d'environ 21% de plus dans l'average end to end delay pour un pause time égale à 200s.

Dans nos travaux futurs, nous prévoyons permettre aux nœuds d'échanger leur

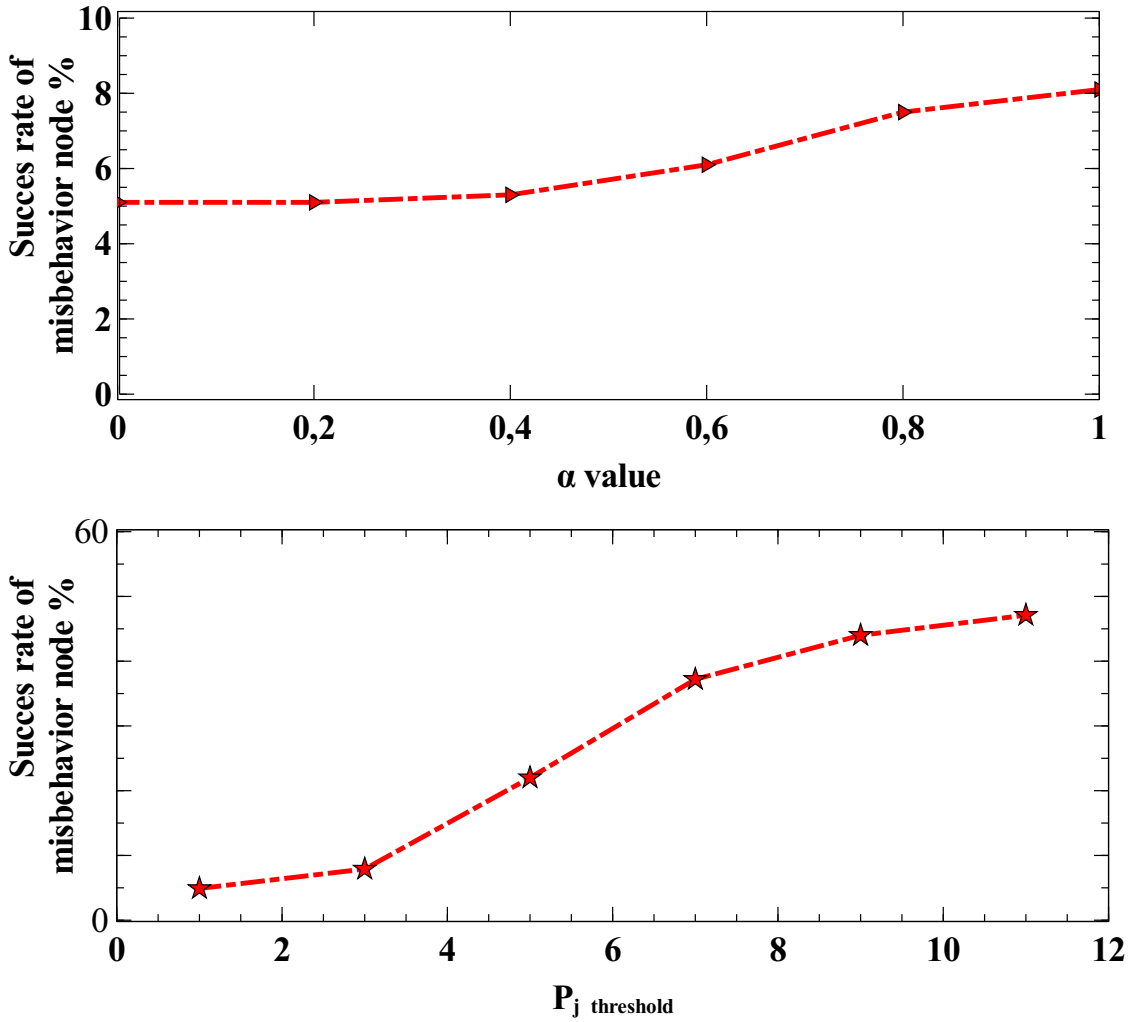

Figure 5.6 – taux de contrôle

réputation et de prendre en compte différents paramètres dans le calcul des valeurs de réputation.

Chapitre 6

Conclusion Générale et perspective

La sécurité dans les réseaux sans fil est un challenge très intéressant, notamment dans les réseaux ad hoc où l'accès au médium physique est totalement libre. Les données transmises par les nœuds peuvent être interceptées ou modifiées. Les protocoles de routage multi-sauts dans les MANETs sont susceptibles á divers types d'attaque suite á la coopération des nœuds et aussi au manque d'une relation de confiance préalable entre eux.

L'attaque de trou noir *black hole* est une attaque active qui affecte le protocole de routage AODV, le nœud malveillant supprime tous les paquets des données détournées vers lui. Les notions présentées dans cette thèse s'articulent autour de ce cadre.

Dans la première partie de cette thèse destinée á la recherche bibliographique, nous avons commencé par une étude globale sur les différents concepts dans les réseaux sans fil, particulièrement toutes les notions du MANETs ainsi que leurs caractéristiques (topologie dynamique, support libre, énergie, mobilité, etc ...).

Nous avons également accordé une part importante pour détailler les services existants qui permettent d'assuré la sécurité tels que (les algorithmes cryptographique, les certificats électroniques, les Fonctions de hachage, les signatures numériques, et les infrastructures de gestion des clés).

Vu la nécessitée du protocole routage, aussi leurs sensibilité et vulnérabilité, nous avons effectué une étude sur les protocoles de routage plate, mono chemin tel que AODV. Nous avons donné un aperçu globale sur leurs classifications selon le groupe de MANETs. les protocoles de routage ad hoc se divisent en trois catégories principales : réactives, proactives et hybrides.

Il est impossible de considérer les mêmes techniques de sécurité pour les différents protocoles de routage ayant des principes de fonctionnement différents et des nécessité de sécurité différents. Ensuite nous avons détaillé le protocole AODV avec

son processus de fonctionnement et ses paquets de contrôle utilisé. Nous avons décrit quelque type d'attaque lié à la couche réseau, en particulier l'attaque de trou noir.

Des solutions différentes de sécurité ont été proposées, il est très difficile de trouver une solution générique pour tous les autres types de routage. En effet, une solution doit présenter un niveau de sécurité contre les nœuds malveillants d'un coté et de garder les performances du protocole d'un autre coté en cours de l'attaque. Dans la deuxième partie de cette thèse nous avons expliqué les deux propositions l'une dans la communication *cross_layer* pour détecter et isoler les nœuds malveillants dans AODV, et l'autre qui est une technique améliorée basée sur la réputation pour détecter les nœuds avec un comportement malveillants dans les MANETs.

Nous avons proposé une technique appelée CrossAODV pour la détection et l'isolation des nœuds malveillants qui utilisent l'attaque de trou noir *black hole* dans le protocole AODV. Nous avons développé une architecture inter-couches *cross_layer* entre la couche MAC et réseau, cette architecture permet á exploiter la fonction de coordination distribuée *DCF*. La solution que nous avons proposé est divisée en deux processus : la vérification et la validation.

Dans la phase de vérification, nous avons anticipé conjointement lors de la découverte de route, via les deux paquets RTS et CTS pour récupérer des informations de routage á propos la destination recherchée. Le processus de validation consiste à confirmer la même information et en comparant les informations de routage demandées avec l'information obtenue durant la phase de vérification. D'après les résultats des simulations que nous avons effectué, nous remarquons une très grande amélioration en terme de PDR et taux de contrôle du protocole sous attaque avec la solution que nous avons proposé.

Notre deuxième contribution consiste a proposé une amélioration d'un système basé sur la réputation pour détecter et isoler le nœud de mauvais comportement dans un réseau mobile ad hoc. Nous avons divisé notre système en trois phases (la surveillance, le calcul de réputation et l'isolation), effectivement on a utilisé seulement la réputation directe *first-hand experience*, où chaque nœud maintien une valeur de réputation pour chacun de ses voisins suite aux paquets transmis.

Selon le nombre prédéfini des paquets envoyés, le nœud émetteur peut calculer la valeur de réputation. Nous avons introduit une amélioration, où la valeur de réputation doit contenir d'autre types de paquets qui sont les paquets d'erreur générés suite à d'autres cas sachant la surcharge de la file d'attente et l'indisponibilité des sauts suivants. Cette amélioration permet de distinguer entre les paquets supprimés via un nœud de mauvais comportement où suite à d'autres événements.

A la fin, Le nœud avec une réputation négative doit être isolé, après le nœud

lance une réparation de route. Aussi, nous avons mis un système de réparation de route qui repose sur *QoS*, nous avons inclus d'autre paramètre tel que le degré de surcharge de la file d'attente et l'énergie résiduelle pour établir une nouvelle route. Nous avons montré á travers des simulations intensives l'intérêt de notre approche dans la détection et l'isolation du nœud de mauvais comportement.

Comme perspective nous proposerons l'extension de ses deux propositions sur plusieurs points de vue.

Toutefois, nous compterons pour aborder le problème de la coopération des nœuds malveillants trou noir contre AODV. De plus nous pourrions adapter notre proposition crossAODV pour les protocoles de routage multi chemins tel que AOMDV.

Autrement, le passage à grand échelle est un autre point ă traiter car une solution avec un nombre des nœuds petit, mais peut avoir un autre comportement dans le coté de performance si on passe à un nombre très grand.

Enfin, Pour donner plus de validité aux résultats que nous avons obtenus par des simulations, il est souhaité de faire des expérimentation avec des équipements réels pour voir exactement leurs fonctionnements.

Bibliographie

Abdelhaq, M., Serhan, S., Alsaqour, R., and Hassan, R. (2011). A local intrusion detection routing security over manet network. In *International Conference on Electrical Engineering and Informatics (ICEEI)*, pages 1–6.

Akhtar, A. K. and Sahoo, G. (2013). Classification of selfish and regular nodes based on reputation values in manet using adaptive decision boundary. *Communications and Network*, 05(3) :7.

Akkaya, K. and Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3) :325–349.

Arya, K. and Rajput, S. (2014). Securing aodv routing protocol in manet using nmac with hbks technique. In *Signal Processing and Integrated Networks (SPIN), 2014 International Conference on*, pages 281–285.

Awerbuch, B., Bar-Noy, A., and Gopal, M. (1994). Approximate distributed bellman-ford algorithms. *IEEE Transactions on Communications*, 42(8) :2515–2517.

Ayday, E. and Fekri, F. (2012). Bp-p2p : Belief propagation-based trust and reputation management for p2p networks. In *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 578–586. IEEE.

Azza, M., Boukli Hacene, S., and Faraoun, k. M. (2015). A cross layer for detection and ignoring black hole attack in manet. *International Journal of Computer Network and Information Security(IJCNIS)*, 7(10) :42–49.

Azza, M., Faraoun, k. M., and Boukli Hacene, S. (2014). A mechanism for detection and ignoring black hole attacker in manet. *The International Conference on Performance Evaluation and Modelling in Wired and Wireless Networks(PEMWN) November*.

Bansal, S. and Baker, M. (2003). Observation-based cooperation enforcement in ad hoc networks. *arXiv preprint cs/0307012*.

Beijar, N. (2002). Zone routing protocol (zrp). *Networking Laboratory, Helsinki University of Technology, Finland*, pages 1–12.

Bettstetter, C., Resta, G., and Santi, P. (2003). The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3) :257–269.

Bouatay, O. (2010). Docteur de l'école supérieure des communications de tunis.

Buchegger, S. and Le Boudec, J.-Y. (2002). Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing*, pages 226–236. ACM.

Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and Thayer, R. (2007). Openpgp message format. Technical report.

Castelluccia, C., Saxena, N., and Yi, J. H. (2007). Robust self-keying mobile ad hoc networks. *Computer Networks*, 51(4) :1169–1182.

Chakeres, I. D. and Belding-Royer, E. M. (2004). Aodv routing protocol implementation design. In *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, pages 698–703. IEEE.

Corson, S. and Macker, J. (IETF Juin 1999). Mobile ad hoc networking (manet). *Technical Report RFC 2501*.

DCF (2004). Le distributed coordination function (dcf). http://www.pouf.org/documentation/securite/html/node17.html, consulté le 21/01/2017.

Desilva, S. and Boppana, R. V. (2005). Mitigating malicious control packet floods in ad hoc networks. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 4, pages 2112–2117. IEEE.

Dini, G. and Duca, A. L. (2012). Towards a reputation-based routing protocol to contrast blackholes in a delay tolerant network. *Ad Hoc Networks*, 10(7) :1167–1178.

Fortz, B. and Thorup, M. (2000). Internet traffic engineering by optimizing ospf weights. In *INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*, volume 2, pages 519–528. IEEE.

Gagandeep, A. and Kumar, P. (2012). Analysis of different security attacks in manets on protocol stack a-review. *International Journal of Engineering and Advanced Technology (IJEAT)*, 1(5) :269–75.

Galice, S. (2007). *Modèle dynamique de sécurité pour réseaux spontanés*. PhD thesis, INSA de Lyon.

Gayraud, V., Nuaymi, L., Dupont, F., Gombault, S., and Tharon, B. (2003). La sécurité dans les réseaux sans fil ad hoc. In *Symposium sur la Sécurité des Téchnologies de l'Information et de la Communication SSTIC, Rennes France*.

Ghosh, U. and Datta, R. (2012). A novel signature scheme to secure distributed dynamic address configuration protocol in mobile ad hoc networks. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2700–2705. IEEE.

Giuseppe, B. and Ilenia, T. (2005). Remarks on ieee 802.11 dcf performance analysis. *Communications Letters, IEEE*, 9(8) :765–767.

Gong, W., You, Z., Chen, D., Zhao, X., Gu, M., and Lam, K.-Y. (2010). Trust based routing for misbehavior detection in ad hoc networks. *Journal of Networks*, 5(5) :551–558.

Haas, Z. J., Pearlman, M. R., and Samar, P. (2002). The zone routing protocol (zrp) for ad hoc networks. *draft-ietf-manet-zone-zrp-04. txt*.

Hajami, A. (2011). *Sécurité du routage dans les réseaux sans fil spontanés : Cas du protocole OLSR*. PhD thesis.

Han, G., Jiang, J., Shu, L., Niu, J., and Chao, H.-C. (2014). Management and applications of trust in wireless sensor networks : A survey. *Journal of Computer and System Sciences*, 80(3) :602–617.

Harmandeep, S. and Manpreet, S. (2013). Securing manets routing protocol under black hole attack. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(4).

Housley, R., Polk, W., Ford, W., and Solo, D. (2002). Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile. Technical report.

Hu, Y.-C., Perrig, A., and Johnson, D. B. (2005). Ariadne : A secure on-demand routing protocol for ad hoc networks. *Wireless networks*, 11(1-2) :21–38.

Hu, Y.-C., Perrig, A., and Johnson, D. B. (2006). Wormhole attacks in wireless networks. *IEEE journal on selected areas in communications*, 24(2) :370–380.

Ibriq, J. and Mahgoub, I. (2004). Cluster-based routing in wireless sensor networks : issues and challenges. In *SPECTS*, volume 4, pages 759–766.

Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., and Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 62–68. IEEE.

Jain, S. and Khunteta, A. (2015). Detection techniques of blackhole attack in mobile adhoc network : A survey. In *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*, ICARCSET '15, pages 47 :1–47 :5, New York, NY, USA. ACM.

Jaiswal, P. and Kumar, R. (2012). Prevention of black hole attack in manet. *International Journal of Computer Networks and Wireless Communications (IJCNWC)*, 2(5).

Jhaveri, R., Patel, S., and Jinwala, D. (2012). A novel approach for grayhole and blackhole attacks in mobile ad hoc networks. In *Advanced Computing Communication Technologies (ACCT), 2012 Second International Conference on*, pages 556–560.

Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., and Degermark, M. (1999). Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 195–206. ACM.

Kaixin, X., Gerla, M., and Sang, B. (2002). How effective is the ieee 802.11 rts/cts handshake in ad hoc networks. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 1, pages 72–76 vol.1.

Kamarularifin, A. J., Zaid, A., and Jamalul-Lail, A. M. (2011). Mitigation of black hole attacks for aodv routing protocol. *International Journal on New Computer Architectures and Their Applications (IJNCAA)*, 1(2).

Kamini, M. and Divakar, S. (2012). Black hole effect analysis and prevention through ids in manet environment. *European Journal of Applied engineering Scienctific Research*, 1(4) :84–90.

Kannhavong, B., Nakayama, H., Nemoto, Y., Kato, N., and Jamalipour, A. (2007). A survey of routing attacks in mobile ad hoc networks. *IEEE Wireless Communications*, 14(5) :85–91.

Karp, B. and Kung, H.-T. (2000). Gpsr : Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM.

khamayseh, Y., Abdulraheem, B., Wail, M., and Muneer, B. (2011). A new protocol for detecting black hole nodes in ad hoc networks. *International Journal of Communication Networks and Information Security (IJCNIS)*, 3(1) :36–47.

Khan, S. and Gupta, V. (2012). A trusted vector method for black hole attack prevention on manet. *International Journal of Computer Science and Management Research*, 1(4).

Kurosawa, S., Nakayama, H., Kato, N., Jamalipour, A., and Nemoto, Y. (2007). Detecting blackhole attack on aodv-based mobile ad hoc networks by dynamic learning method. *IJ Network Security*, 5(3) :338–346.

Lalit, H., Vishal, V., and Nagesh, C. (May 2011). Preventing aodv routing protocol from black hole attack. *International Journal of Engineering Science and Technology (IJEST)*, 3(5).

Lou, W. and Fang, Y. (2004). A survey of wireless security in mobile ad hoc networks : challenges and available solutions. In *Ad hoc wireless networking*, pages 319–364. Springer.

Marchang, N. and Datta, R. (2012). Light-weight trust-based routing protocol for mobile ad hoc networks. *Information Security, IET*, 6(2) :77–83.

Marina, M. K. and Das, S. R. (2001). On-demand multipath distance vector routing in ad hoc networks. In *Ninth International Conference on Network Procotols (ICNP)*, pages 14–23. IEEE.

Marti, S., Giuli, T. J., Lai, K., and Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265. ACM.

Michiardi, P. and Molva, R. (2002). *Core : a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks*, pages 107–121. Springer.

Ming-Yang, S., Kun-Lin, C., and Wei-Cheng, L. (2010). Mitigation of black-hole nodes in mobile ad hoc networks. In *Parallel and Distributed Processing with Applications (ISPA), 2010 International Symposium on*, pages 162–167.

Mittal, S. and Kaur, P. (2009). Performance comparison of aodv, dsr and zrp routing protocols in manet's. In *Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on*, pages 165–168. IEEE.

Nital, M., Devesh, C. J., and Mukesh, Z. (2010). Improving aodv protocol against blackhole attacks. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2.

Patel, A. D. and Jhaveri, R. H. (2016). Addressing packet forwarding misbehavior with two phase security scheme for aodv-based manets. *International Journal of Computer Network and Information Security*, 8(5).

Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad hoc on-demand distance vector (aodv) routing.

Perkins, C. E. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *ACM SIGCOMM computer communication review*, volume 24, pages 234–244. ACM.

Perkins, C. E. and Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. In *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, WMCSA '99, pages 90–, Washington, DC, USA. IEEE Computer Society.

Rai, A. K., Tewari, R. R., and Upadhyay, S. K. (2010). Different types of attacks on integrated manet-internet communication. *International Journal of Computer Science and Security*, 4(3) :265–274.

Raj, P. N. and Swadas, P. B. (2009). Dpraodv : A dyanamic learning system against blackhole attack in aodv based manet. *International Journal of Computer Science Issues*, abs/0909.2371.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2) :120–126.

Rogaway, P. and Shrimpton, T. (2004). Cryptographic hash-function basics : Definitions, implications, and separations for preimage resistance, second-preimage

resistance, and collision resistance. In *International Workshop on Fast Software Encryption*, pages 371–388. Springer.

Sachan, P. and Khilar, P. M. (2011). Securing aodv routing protocol in manet based on cryptographic authentication mechanism. *International Journal of Network Security & Its Applications*, 3(5) :229.

Sanzgiri, K., Dahill, B., Levine, B. N., Shields, C., and Belding-Royer, E. M. (2002). A secure routing protocol for ad hoc networks. In *Proceedings. 10th IEEE International Conference on Network Protocols*, pages 78–87. IEEE.

SB, M. K. and Benni, N. K. S. (2013). Cryptographic approach to overcome black hole attack in manets. *International Journal of Innovations in Engineering and Technology (IJIET)*, 2(3) :86–92.

Senthilkumar, S. and William, J. (2014). A survey on reputation based selfish node detection techniques in mobile ad hoc network. *Journal of Theoretical & Applied Information Technology*, 60(2).

Shakkottai, S., Rappaport, T. S., and Karlsson, P. C. (2003). Cross-layer design for wireless networks. *IEEE Communications magazine*, 41(10) :74–80.

Singh, M. and Kaur, G. (2013). A surveys of attacks in manet. *International Journal of Advanced Research in Computer Sciences and Software Engineering (IJARCSSE)*, 3(6).

Singh, S., Sharma, S., and Sahu, S. (2012). Secure aodv using symmetric key cryptography with cyclic chain hash function (cchf). *International Journal of Computer Applications*, 47(18).

Sonal, K. N. (2013). Black hole attack detection using fuzzy logic. *International Journal of Science and Research (IJSR)*, 2(8).

Sowmya, K. S. and Mayuri, G. (2013). Prevention of black hole attack in secure routing protocol. *International Journal at International Journal of Science (IJSR)*, 2(6).

Srivastava, V. and Motani, M. (2005). Cross-layer design : a survey and the road ahead. *IEEE Communications Magazine*, 43(12) :112–119.

Subash, C. M. and Surya, N. P. (2011). A counter measure to black hole attack on aodv based mobile ad-hoc networks. *International Journal of Computer and Communication Technology (IJCCT)*.

Tan, S. and Keecheon, K. (2013). Secure route discovery for preventing black hole attacks on aodv-based manets. In *ICT Convergence (ICTC), 2013 International Conference on*, pages 1027–1032.

Tuteja, A., Gujral, R., and Thalia, S. (2010). Comparative performance analysis of dsdv, aodv and dsr routing protocols in manet using ns2. In *Advances in Computer Engineering (ACE), 2010 International Conference on*, pages 330–333. IEEE.

Vani, A. and Sreenivasa Rao, D. (2011). Removal of black hole attack in ad hoc wireless networks to provide confidentiality security service. *International Journal of Engineering Science and Technology (IJEST)*, 3(3).

Varshney, T., Sharma, T., and Sharma, P. (2014). Implementation of watchdog protocol with aodv in mobile ad hoc network. In *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*, pages 217–221.

Velloso, P., Laufer, R., de O Cunha, D., Duarte, O., and Pujolle, G. (2010). Trust management in mobile ad hoc networks using a scalable maturity-based model. *Network and Service Management, IEEE Transactions on*, 7(3) :172–185.

Wang, F., Wang, F., Huang, B., and Yang, L. T. (2010). Cosr : a reputation-based secure route protocol in manet. *EURASIP J. Wirel. Commun. Netw.*, 2010 :1–13.

Wu, B., Chen, J., Wu, J., and Cardei, M. (2007). A survey of attacks and countermeasures in mobile ad hoc networks. In *Wireless Network Security*, pages 103–135. Springer.

Yang, X. and Rosdahl, J. (2002). Throughput and delay limits of ieee 802.11. *Communications Letters, IEEE*, 6(8) :355–357.

Yerneni, R. and Sarje, A. K. (2012). Enhancing performance of aodv against black hole attack. In *Proceedings of the CUBE International Information Technology Conference*, pages 857–862, New York, NY, USA. ACM.

Yi, P., Dai, Z., Zhong, Y., and Zhang, S. (2005). Resisting flooding attacks in ad hoc networks. In *International Conference on Information Technology : Coding and Computing (ITCC'05)-Volume II*, volume 2, pages 657–662. IEEE.

Zhen, J. and Srinivas, S. (2003). Preventing replay attacks for secure routing in ad hoc networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 140–150. Springer.

LISTE DES PUBLICATIONS

Journaux avec comité de lecture

Mohammed Azza, Sofiane Boukli Hacene, and kamel Mohamed Faraoun. A cross layer for detection and ignoring black hole attack in manet. I. J. Computer Network and Information Security, 7(10) pp : 42-49, September 2015. DOI : 10.5815/ijcnis.2015.10.05

Mohammed Azza and Sofiane Boukli Hacene. An Enhanced reputation-based for Detecting Misbehaving nodes in MANET. soumis. International Journal of Wireless and Microwave Technologies(IJWMT).

Conférences internationales

Azza Mohammed, Faraoun Kamel Mohamed and Sofiane Boukli Hacene. A Mechanism for Detection and Ignoring Black Hole Attacker in MANET.The International Conference on Performance Evaluation and Modelling in Wired and Wireless Networks(PEMWN) November, Sousse, Tunisia,2014.

Annexe A

Environnement de simulations (Network Simulator 2)

A.1 Présentation de Network Simulator 2

NS2 (Network Simulator 2) est un simulateur de réseau développé pour faire des recherche, il est basé sur les évènements discrets. C'est un environnement riche et populaire, permet de réaliser des simulations du différents protocoles d'IP dans des environnements filaires et sans fil. Le simulateur utilise le langage orienté objet OTCL dérivé de TCL pour la description des topologie de simulation sous forme d'un script.

NS-2, est l'un des simulateurs réseau qui est largement utilisé par la communauté scientifique des réseaux. Il a été crée par l'université de Berkeley, USC (University of Southern California) et Xerox PARC dan le cadre du projet VINT (Virtual Inter Network Testbed). Ce projet est validé par le DARPA (Defense Advanced Research Projects Agency). NS-2 est un outil de recherche, il sert aussi bien dans l'étude des protocoles de routage des réseaux mobiles ou les communications par satellites, il permet à l'utilisateur de définir un réseau et de simuler les communications entre les nœuds.

le script contient une description globale sur la topologie du réseau, les caractéristiques des liens physiques, les protocoles utilisés, le type de trafic généré par les sources, les événements, etc.... Le corps de simulateur est écrit en c++, cela donne une puissance d'exécution et calcul aux différents protocoles. Le résultat d'une simulation est un fichier trace contenant tous les événements de la simulation. À travers un traitement on extrait toutes les informations désirées. Par ailleurs, le simulateur

permet la création d'un fichier d'animation (d'extension .nam), permettant de visualiser la simulation sur l'interface graphique NAM. Cette visualisation fournit une représentation du graphique du réseau sur laquelle on peut voir les paquets circulent, suivre le niveau des files d'attente et observer le débit courant des liaisons. La figure A.1 présente une simple utilisation de NS-2.

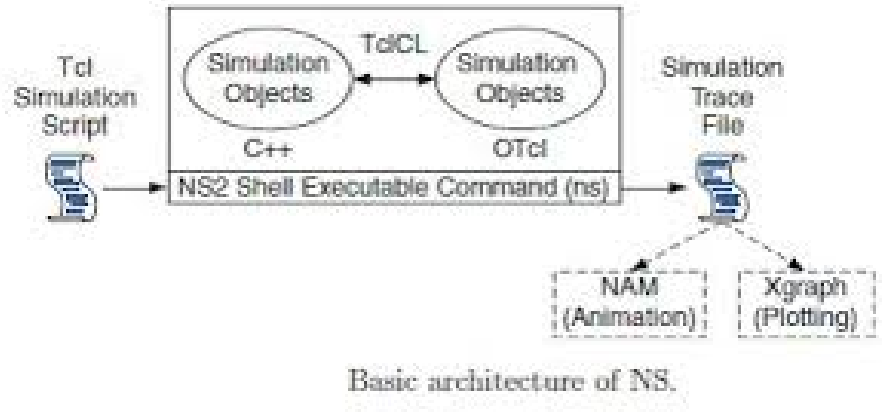

Figure A.1 – simple utilisation de NS-2

NS-2 est conçu initialement pour fonctionner sur les systèmes d'exploitation Unix et Linux, mais il existe un moyen pour son installation sur un système Windows. Le simulateur NS-2 est fourni sous forme d'un paquetage qui regroupe tous les fichiers nécessaires à son installation. Plusieurs principaux composants sont actuellement disponible dans NS-2, nous les représentons par catégorie dans le tableau suivant :

Table A.1 – Liste des composants dans Ns-2

Application	Web, ftp, telnet, générateur de trafic (CBR).
Transport	TCP, UDP, RTP, SRM.
Routage	OLSR, DSR, AODV.
Gestion de file d'attente	RED, DropTail, Token bucket.
Discipline de service	CBQ, SFQ, DRR, Fair queueing.
Système de transmission	CSMA/CD, CSMA/CA, P2P.

A.2 Architecture de NS-2

Arborescence des classes compilées

Un grand nombre de classes sont prédéfinies et mettent en œuvre plusieurs types de protocoles, de files d'attentes, de sources et algorithmes de routage.

TclObject : C'est la racine de toutes les autres classes à la fois dans l'arborescence compilée et interprétée.

NsObject : c'est une sous-classe de la classe TclObject mais reste cependant une superclasse aux autres classes. La principale différence avec la classe TclObject tient à ce qu'elle soit capable de recevoir des paquets et traiter des événements. Elle est réellement définie par les sous-classes :

Application : Classe mère de toutes les applications (ftp, telnet, web).

Agent : La classe agent fournit des méthodes utiles au développement de la couche transport et à d'autres protocoles du plan de signalisation ou de gestion. C'est la classe de base pour définir des nouveaux protocoles dans NS-2. .

Node : un nœud peut être une machine hôte, un switch, un routeur, une passerelle, etc. Chaque nœud contient au minimum les composants suivants :

1. Une adresse ou un identificateur (id_) automatiquement incrémenté par une unité (à partir de 0) quand les nœuds sont créés.
2. Une liste de nœuds voisins (neighbor_).
3. Une liste d'agents (agent_).
4. Un identificateur du type du nœud (nodetype_).
5. Un module de routage.

Queue : la classe mère de tous les buffers (DropTail, RED) LinkDelay : cette classe simule le délai de propagation et le temps de transmission sur les liens du réseau. Avec la classe Queue, cette classe simule les couches 1 et 2 .

Packet : la classe de tous les paquets circulant sur le réseau.

TimerHundler : la classe mère de tous les timers (temporisateurs) utilisés par les protocoles du réseau.

Arborescence des fichiers

La distribution de NS-2 comprend principalement 3 répertoires :

– ns-2 : l'application NS. Ce répertoire contient l'ensemble des fichiers .h et .cc de NS-2.

– nam-1 : l'outil de visualisation des résultats de la simulation : l'animateur réseau.
– tclcl : codes sources assurant la liaison entre l'interpréteur et le simulateur. L'un des principaux fichiers est : tcl-object.tcl.

A.3 Le modèle réseau sous NS-2

Un modèle réseau sous NS-2 est constitué de quatre composants essentiels :
– Les nœuds du réseau : endroits où est généré le trafic, ou nœuds de routage
– Les liens de communication entre les nœuds
– Les agents de communication ces agents sont attachés aux nœuds et connectés l'un à l'autre, ce qui représente un échange de données (connexion TCP, flux UDP).
– Les applications qui génèrent le trafic de données selon certaines lois (CBR, VBR), et se servent des agents de transport.

A.4 Traitement des résultats dans NS-2

Après le déroulement de la simulation, NS-2 génère une trace sous forme d'un fichier texte contenant tous les événements de la simulation.

Chaque événement est représenté dans le fichier trace avec une ligne qui contient douze champs. Le tableau A.2 donne une vision de la structure d'une ligne du fichier trace sous NS-2

Table A.2 – Structure d'une ligne du fichier trace.

1	2	3	4	5	6	7	8	9	10	11	12
Event	Time	From node	To node	Pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id

1. Action effectuée sur le paquet. Un '+' signifie que le paquet est reçu dans une file, un '-' signifie que le paquet quitte la file, un 's' signifie que le paquet est envoyé, un 'd' signifie que le paquet est jeté (dropé) et un 'r' signifie que le paquet est réceptionné par un agent.
2. Instant où l'action est effectuée.
3. Nœud de départ du lien concerné.
4. Nœud d'arrivée du lien concerné.
5. Type de paquet.

6. Taille du paquet en bytes.
7. Flags.
8. Identificateur de flux.
9. Agent de départ.
10. Agent d'arrivée.
11. Numéro de séquence.
12. Identificateur unique pour chaque paquet.

En plus des fichiers traces qu'offre le simulateur NS-2, il permet de visualiser les événements de la simulation à travers une interface graphique.

A.5 Les différents modèles de mobilité sous NS-2

Le déroulement de chaque simulation des réseaux ad hoc (MANET) est lié a leurs paramètres spécifie au préalable, ainsi, l'évaluation d'un protocole de routage ad hoc dépend du choix d'un modèle de mobilité pour visionner les mouvements réalistes des nœuds. Il existe plusieurs type de modèles de mobilité, certains catégories représentent les nœuds mobiles dont les mouvements sont indépendants l'un de l'autre. D'autre part, il y a des modèles de mobilité de groupe qui représentent les nœuds mobiles dont les mouvements dépendent les uns des autres et ils tendent à être plus réalistes dans les applications impliquant la communication de groupe.

Le modèle Radom Way Point (RWP)

Dans ce modèle la mobilité des nœuds est typiquement aléatoire et tous les nœuds sont distribués uniformément dans l'espace de simulation. En effet il consiste à :

- Le placement d'un certain nombre de mobiles dans une zone rectangulaire de laquelle ils ne peuvent sortir.
- L'affectation d'une position, d'une vitesse et d'une destination initiale à chaque mobile.
- Chaque fois que les mobiles atteignent leur destination dans le carré, ils déplacement vers une autre destination choisie aléatoirement après un éventuel temps de pause.

Le modèle Random Walk

Ce modèle est développé pour imiter un mouvement inattendu. Un nœud mobile dans ce modèle se déplace de son endroit actuel à un nouvel endroit en choisissant

aléatoirement une direction et une vitesse suivant lesquelles il se déplace. Les nouvelles vitesse et direction sont choisies dans des gammes prédéfinies, respectivement $[speedmin, speedmax]$ et $[\theta, 2\pi]$. Un nœud mobile atteignant la frontière de simulation, rebond avec l'angle déterminé par la direction entrante et puis continue le long du nouveau chemin.

Le modèle aléatoire de direction (random direction model)

Ce modèle essaie d'alléger ce comportement, fournissant un nombre constant de voisins dans toute la simulation. Les nœuds mobiles choisissent une direction aléatoire suivant laquelle ils se déplacent en tant que modèle de mobilité de random walk, où ils se déplacent vers la frontière de la simulation dans cette direction. Une fois que la frontière est atteinte, le nœud mobile fait une pause pendant le temps indiqué, choisit une autre direction angulaire entre (0 et 180) et continue alors le processus.

Annexe B

Script de simulations

```
#===================================
# Parametros de la linea de comandos
#===================================
global argv arg0

set opt(tr) [lindex $argv 0];# fichier trace
set opt(na) [lindex $argv 1];# fichier nam
set opt(cp) [lindex $argv 2];#scenarios de traffic
set opt(sc) [lindex $argv 3];# scenarios de mobilite

puts "---------------------------------"
puts $opt(tr)
puts $opt(na)
puts $opt(cp)
puts $opt(sc)
puts "---------------------------------"

# Define options
# Define options
#===================================================
set opt(chan)    Channel/WirelessChannel
set opt(prop)    Propagation/TwoRayGround
set opt(netif)   Phy/WirelessPhy
set opt(mac)     Mac/802_11
set opt(ifq)     Queue/DropTail/PriQueue
set opt(ll)              LL
```

```
set opt(ant)         Antenna/OmniAntenna
set opt(x)      500   ;# X dimension of the topography
set opt(y)      500   ;# Y dimension of the topography
set opt(ifqlen) 50         ;# max packet in ifq
set opt(seed)   0.0
set opt(adhocRouting)   AODV
set opt(nn)     50   ;# how many nodes are simulated

set opt(stop) 200         ;# simulation time
#===================================
# Main Program
#===================================
# Initialize Global Variables
# create simulator instance
set ns_         [new Simulator]

# set wireless channel, radio-model and topography objects
set wtopo       [new Topography]

# create trace object for ns and nam
set tracefd     [open $opt(tr) w]
$ns_ trace-all $tracefd
set namtrace      [open $opt(na) w]
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)
# use new trace file format
$ns_ use-newtrace

# define topology
$wtopo load_flatgrid $opt(x) $opt(y)

# Create God
set god_ [create-god 60]

# define how node should be created
#global node setting
$ns_ node-config -adhocRouting ipsAODV \
-llType $opt(ll) \
```

```
-macType $opt(mac) \
-ifqType $opt(ifq) \
-ifqLen $opt(ifqlen) \
-antType $opt(ant) \
-propType $opt(prop) \
-phyType $opt(netif) \
-channelType $opt(chan) \
-topoInstance $wtopo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON

#  Create the specified number of nodes [$opt(nn)]
# and "attach" them to the channel.
for {set i 0} {$i < $opt(nn)+1 } {incr i} {
        set node_($i) [$ns_ node]
        $node_($i) random-motion 0;# disable random motion
        puts "routing access from mac f"
        set nmac_($i) [$node_($i) set mac_(0)]
        set naodv_($i) [$node_($i) agent 255]
        $nmac_($i) access-aodv $naodv_($i)

        $naodv_($i) access-mac $nmac_($i)

}
#source $opt(b)
$ns_ node-config  -adhocRouting blackAODV
set node_(51) [$ns_ node]
$node_(51) set X_ 250
$node_(51) set Y_ 200
$node_(51) color red
$ns_ at 0.0 "$node_(51) label \"blackhole node\""
$ns_ at 0.0 "$node_(51) color red"

# Define node movement model
puts "Loading connection pattern..."
source $opt(cp)
```

```
# Define traffic model
puts "Loading scenario file..."
source $opt(sc)

# Define node initial position in nam
for {set i 0} {$i < $opt(nn)+2} {incr i} {

# 20 defines the node size in nam,
# must adjust it according to your scenario
# The function must called after mobility model is defined
        $ns_ initial_node_pos $node_($i) 20

}

# Tell nodes when the simulation ends
for {set i 0} {$i < $opt(nn)+2 } {incr i} {
        $ns_ at $opt(stop).000000001 "$node_($i) reset";
}

$ns_ at $opt(stop) "stop"
proc stop {} {
        global ns_ tracefd namtrace
        $ns_ flush-trace
        close $tracefd
        close $namtrace
        #exec nam $opt(na) &
        exit 0
}
$ns_ run
```

Code B.1 – Fichier cross.tcl

```
###----------batch azza----------####
#!/bin/csh

unset noclobber
set outdir = senario
```

```
##aodv normal
set outdir1 = rep1
##aodv avec attaque
set outdir2 = rep2
##aodv acec crosslayer
set outdir3 = rep3
set outdir4 = rep4
set maxspeed = 10
set numnodes = 50
set maxx = 500
set maxy = 500
set simtime = 200

foreach scen (10 25 40)
foreach pt (0 50 100 150 200)

#####----------aodv cross detection----------#####
ns bsimple.tcl  $outdir3/resltscross${scen}-${pt}.tr
$outdir3/resltscross${scen}-${pt}.nam $outdir/cbr${scen}
$outdir/sena50-${pt}

awk -f pdr.awk $outdir3/resltscross${scen}-${pt}.tr x=${pt}
>> $outdir4/PDRcross${scen}

awk -f delay.awk $outdir3/resltscross${scen}-${pt}.tr y=${pt}
 >> $outdir4/ENDcross${scen}
awk -f overhead.awk $outdir3/resltscross${scen}-${pt}.tr
z=${pt} >> $outdir4/OVERcross${scen}
######----------aodv normal----------#######

ns simple.tcl  $outdir3/resltsn${scen}-${pt}.tr
$outdir3/resltsn${scen}-${pt}.nam $outdir/cbr${scen}
$outdir/sena50-${pt}

awk -f pdr.awk $outdir3/resltsn${scen}-${pt}.tr x=${pt}
>> $outdir4/PDRn${scen}
```

```
awk -f delay.awk $outdir3/resltsn${scen}-${pt}.tr y=${pt}
>> $outdir4/ENDn${scen}
awk -f overhead.awk $outdir3/resltsn${scen}-${pt}.tr z=${pt}
 >> $outdir4/OVERn${scen}

#####--------------aodv avec black-------------##########
ns simpleblack.tcl  $outdir3/resltsb${scen}-${pt}.tr
$outdir3/resltsb${scen}-${pt}.nam $outdir/cbr${scen}
$outdir/sena50-${pt}

awk -f pdr.awk $outdir3/resltsb${scen}-${pt}.tr x=${pt}
>> $outdir4/PDRb${scen}
awk -f delay.awk $outdir3/resltsb${scen}-${pt}.tr y=${pt}
>> $outdir4/ENDb${scen}
awk -f overhead.awk $outdir3/resltsb${scen}-${pt}.tr
z=${pt} >> $outdir4/OVERb${scen}
end
end
```

Code B.2 – Fichier azza.csh

```
##################################################################
#       AWK Script to calculate PDR Packet Delivator Ratio#
##################################################################
BEGIN {
sendLine = 0;
recvLine = 0;
fowardLine = 0;
}

$0 ~/^s.* AGT/ {
sendLine ++ ;
}

$0 ~/^r.* AGT/ {
recvLine ++ ;
}
```

```
$0 ~/^f.* RTR/ {
fowardLine ++ ;
}

END {
if (ARGV[2]=="x=0")      print 100*(recvLine/sendLine);
if (ARGV[2]=="x=50")    print 100*(recvLine/sendLine);
if (ARGV[2]=="x=100")    print 100*(recvLine/sendLine);
if (ARGV[2]=="x=150")    print 100*(recvLine/sendLine);
if (ARGV[2]=="x=200")    print 100*(recvLine/sendLine);

}
```

Code B.3 – Fichier pdr.awk

```
##############################################################################
#    AWK Script to calculate Normalized Routing Load      #
##############################################################################

BEGIN{
recvd = 0;#to calculate data packets received
rt_pkts = 0;#to calculate routing packets received
}

{
##### Check if it is a data packet
if (( $1 == "r") && ( $35 == "cbr" || $35 =="tcp" )
&& ( $19=="AGT" )) recvd++;

##### Check if it is a routing packet
if (($1 == "s" || $1 == "f") && $19 == "RTR"
 && ($35 =="AODV")) rt_pkts++;
#if ($57 == "ALERT" && $1 == "s") rt_pkts++;
#if ($61 == "REQUEST" ) req++;
#if ($57 == "REPLY" ) rep++;
#if ($57 == "ERROR" ) err++;
}
```

```
END{

if (ARGV[2]=="z=0")       print (rt_pkts/recvd);
if (ARGV[2]=="z=50")      print (rt_pkts/recvd);
if (ARGV[2]=="z=100")     print (rt_pkts/recvd);
if (ARGV[2]=="z=150")     print (rt_pkts/recvd);
if (ARGV[2]=="z=200")     print (rt_pkts/recvd);
}
```

Code B.4 – Fichier overhead.awk :

```
##################################################################
#    AWK Script to calculate Average End-to-End Dela          #
##################################################################

BEGIN {
seqno = -1;
sent = 0;
#     droppedPackets = 0;

#     receivedPkts = 0;
count = 0;
}
{
if($19 == "AGT" && $1 == "s" && seqno < $41
 && ($35 == "cbr")) {

seqno = $41;
start_time[$41] = $3;
#sent++;

}
else if(($19 == "AGT") && ($1 == "r") &&
($35 == "cbr")) {

receivedPkts++;
end_time[$41] = $3;
```

```
}
#else if ($1 == "D" && $7 == "tcp" && $8 > 512){

#             droppedPackets++;
#  }
 else #if($1 == "d" && $35 == "cbr" && $19 == "AGT")
 {
 end_time[$41] = -1;
 }
 }
 END {
 for(i=0; i<=seqno; i++) {
 count = end_time[i] - start_time[i];
 if (count >0) n2ndelay += count;
 }
 if (ARGV[2]=="y=0")   print (n2ndelay/receivedPkts)*1000;
 if (ARGV[2]=="y=50")  print (n2ndelay/receivedPkts)*1000;
 if (ARGV[2]=="y=100") print (n2ndelay/receivedPkts)*1000;
 if (ARGV[2]=="y=150") print (n2ndelay/receivedPkts)*1000;
 if (ARGV[2]=="y=200") print (n2ndelay/receivedPkts)*1000;
 }
```

Code B.5 – Fichier delay.awk

Résumé

Le développement continu des réseaux, avec notamment l'existence de la conception des dispositifs sans fil rend les réseaux ad hoc comme une technologie de plus en plus adoptée. Un réseau ad hoc mobile appelé généralement MANET (Mobile Ad hoc NETwork), est un système autonome des nœuds mobiles reliés par des liens sans fil formant un réseau temporaire à topologie variable, avec un fonctionnement sans station de base et sans administration centralisée.

Dans de tels environnements, les hôtes mobiles sont obligés de se comporter comme des routeurs afin de maintenir les informations de routage du réseau. Les protocoles de routage ad hoc existant dans la littérature font l'hypothèse d'un environnement idéal dans lequel le fonctionnement du réseau n'est pas soumis à des attaques malveillantes. C'est la raison pour laquelle de nombreuses vulnérabilités au niveau du routage sont apparus. Cependant, concevoir des mécanismes de sécurité fiable pour les réseaux ad hoc est un challenge.

Différentes techniques ont été proposées pour sécuriser certains protocoles de routage envisagés pour les réseaux ad hoc. Ce domaine reste très complexe et fertile. Dans ce travail de recherche, nous considérons le problème de la sécurisation des informations de routage du protocole réactive AODV contre l'attaque de trou noire. Suite à l'étude effectuée sur le protocole AODV, ainsi que les approches de sécurité proposées pour le sécuriser, nous avons proposé un schéma de routage sécurisé intercouche et un autre qui se base sur la réputation. Nos approches consistent à intégrer la méthode d'accès au support DCF de la couche MAC dans le protocole AODV et juger le nœud malveillant suite a son réputation.

Pour mesurer nos mécanismes qui seront également proposés pour sécuriser le protocole de routage AODV contre l'attaque de trou noir, nous avons évalué leurs performances à travers plusieurs simulations à travers le simulateur NS2 (Network Simulator). Les résultats obtenus montrent que nos approches ont atteint un équilibre entre les performances du protocole et le niveau de sécurité offert.

Mots clés : MANET, sécurité, AODV, trou noir, Mauvais comportement,réputation.

Abstract

The continued development of networks, including the wireless devices makes ad hoc networks technology more adopted. A mobile ad hoc network called MANET (Mobile Ad hoc NETwork) is an autonomous system of mobile nodes connected by wireless links forming a dynamic topology and a temporary operating network without a base station and without centralized administration.

In these environments, mobiles nodes are forced to act as routers to maintain the routing information in network. Ad hoc routing protocols exist in the literature assume that is an ideal environment which the operation of the network is not subject to malicious attacks. This is the reason why many routing vulnerabilities have appeared. To build a secure and a reliable routing protocol for ad hoc networks is a challenge. Different techniques have been proposed to secure some of the routing protocols envisaged in ad hoc networks. This area of research is very complex and new. In this research work, we consider the problem of securing the reactive protocol AODV against malicious behavior.

After a study on the AODV protocol, as well as the proposed security approaches to secure it, we propose a cross layer secure routing scheme. Our approach consists to integrate the function of access in the support of layer two (DCF) in the AODV protocol. We have proposed another contribution called an enhanced reputation-based method to detect and isolate the misbehavior node in mobile ad hoc networks to suspend the malicious node. Our approach composed of three phases (Monitoring, Computes reputation, Isolation and route maintenance).

To measure our mechanisms that will also be proposed to secure the AODV routing protocol against black hole attack. Their performance was evaluated through several simulations through the simulator NS2 (Network Simulator 2). The results show that our approach reach a balance between the performance of the protocol and the level of security offered.

Keywords: MANET, security, AODV, black hole, Misbehavior, reputation-based.