TABLE DES MATIÈRES

Page

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ANSN	Advertised Neighbor Sequence Number
CF	Cost to Forward
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IETF	The Internet Engineering Task Force
IMS	IP Multimedia Subsystem
LTE	Long Term Evolution
MANET	Mobile Ad-Hoc Network
MIPV6	Mobile IP version 6
MPR	Mulipoint Relay
N1H	One Hop Neighbor
N2H	Two Hop Neighbor
NAM	Network Animator
NS2	Network Simulator version 2
OLSR	Optimized Link State Routing
PKI	Public Key infrastructure
RREP	Route REPly
RREQ	Route REQuest
TC	Topology Control
TCL	Tool Command Language
TTL	Times To live
VANET	Vehicular Ad-hoc NETwork
VoIP	Voice Over IP
WSN	Wireless Sensor Network
ZRP	Zone Routing Protocole

Rapport-Gratuit.com

INTRODUCTION

1) Point de départ

Nous assistons ces dernières années à une révolution technologique remarquable où la communication mobile a pu se démarquer. Les terminaux cellulaires deviennent plus «intelligents» que leurs ancêtres en profitant du développement des domaines technologiques connexes en l'occurrence : l'autonomie des batteries, la puissance des processeurs ainsi qu'une bande passante élevée offrant à ces terminaux plus d'autonomie, de puissance de calcul et de support pour les applications multimédia. Cette révolution a changé le comportement des utilisateurs qui a passé d'un comportement statique et passif à un comportement nomade ou en totale mobilité.

Devant ce développement, l'industrie des télécommunications a dû introduire de nouveaux standards capables de répondre aux nouveaux besoins de ses utilisateurs en matière d'interactivité (LTE/IMS) ou de mobilité (MIPv6). Toutefois l'interopérabilité, la flexibilité et l'auto-configuration nécessitent encore des développements importants : les réseaux Ad-Hoc s'avèrent une solution capable de répondre à ces besoins.

Un réseau Ad-Hoc est un réseau sans fil constitué de plusieurs terminaux mobiles (PC portable, tablette, PDA…). L'échange de données entre ces terminaux se fait sans l'existence d'une infrastructure dédiée. Seul un protocole de routage est exigé pour gérer les transmissions de données, offrant ainsi une flexibilité et une auto-configuration à ses utilisateurs. La nature décentralisée de ce type de réseau le qualifie pour être utilisé pour des applications qui exigent un déploiement rapide d'un réseau de communication lors de désastre ou de guerre. De récentes recherches ont proposé de nouvelles applications des réseaux Ad-Hoc dans le domaine du transport (*Vehicular Ad-Hoc Network*), des capteurs (*Wireless Sensors Network*) ou des applications civiles (*Mobile Ad-hoc Network*). La technologie Ad-Hoc risque de connaitre un grand essor grâce surtout à la création de groupes de recherche et de standardisation comme le groupe MANET.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

La mission du groupe MANET (groupe de recherche pour les MANET à l'IETF) consiste à standardiser les protocoles de routage qui se chargent d'acheminer les données entre les terminaux hors portée radio. Malheureusement la plupart de ces protocoles n'ont pas été conçus en considérant l'un des points primordiaux dans les infrastructures de communication de nos jours à savoir : la sécurité. La confidentialité, l'intégrité et la disponibilité sont devenues des besoins essentiels surtout dans les réseaux Ad-Hoc. La nature versatile et coopérative de ce type de réseaux oblige les concepteurs à proposer des mécanismes pour détecter toute activité malicieuse nuisant au bon fonctionnement du réseau.

2) Objectif de la recherche

Les protocoles de routage MANET sont très diversifiés par leur nature. Leur mécanisme de routage utilisé et leur processus de maintien des routes adoptés sont très variés. Dans ce projet de recherche, nous concentrons nos efforts sur le protocole OLSR (*Optimised Link State Routing*) (Clausen et al., 2003) vu sa popularité. Ce protocole utilise une technique de diffusion de message (*Message Broadcasting*) optimisée en introduisant la notion des nœuds MPR (*Multipoint Relay*). Ces nœuds ont deux rôles : diffuser les informations topologiques du réseau d'une part, et servir à acheminer les paquets vers leurs destinations finales d'autre part. Ces nœuds sont considérés comme des représentants des destinations en question. Tout autre nœud qui veut rejoindre un nœud destination doit passer par un des MPR de ce nœud.

Comme dans la plupart des protocoles MANET, aucun mécanisme de sécurité n'a pas été pris en considération lors de la conception du protocole OLSR (Adjih et al., 2003), ce qui l'expose à plusieurs sortes d'attaques visant la transmission des données ou le protocole de routage proprement dit. Le processus d'acheminement des données de la source vers la destination est un processus collaboratif pendant lequel les nœuds devraient engager leurs ressources afin de l'accomplir. Toutefois, certains nœuds expriment un certain égoïsme en refusant de participer à ce processus. SC-OLSR (Chriqi, Otrok et Robert, 2009) a été créé dans l'optique d'encourager les nœuds à participer honnêtement à ce processus moyennant

des mécanismes d'incitation et d'encouragement. SC-OLSR a aussi pour but de prolonger la durée de vie des nœuds en modifiant leur organisation aléatoire (dans le classique OLSR) à une organisation sous forme de grappes (ou *Clusters*). Chaque grappe est représentée par un chef de grappe (ou *Cluster Head* (CH) – pour le reste de ce mémoire, l'acronyme anglaise CH sera utilisée) imitant ainsi les architectures centralisées des réseaux filaires.

Le protocole SC-OLSR a pu atteindre ses objectifs en matière d'optimisation de ressources utilisées. Toutefois, il souffre toujours de certaines vulnérabilités qui pourraient nuire au bon fonctionnement de ce protocole. Notre objectif consiste donc à caractériser ces vulnérabilités, établir un modèle d'attaque pour les nœuds malicieux potentiels et implanter les mécanismes nécessaires pour détecter et punir ces nœuds sous forme de système de détection d'intrusion (*Intrusion Detection System* (IDS)).

L'objectif de ce projet de recherche est de déterminer l'efficacité d'un système de détection d'intrusion distribué. Nous devons ainsi déterminer si un nombre limité de nœuds de surveillance est suffisant pour obtenir un niveau de sécurité acceptable. Pour déterminer l'efficacité de notre solution, nous allons la comparer à la solution *Watchdog* (Marti et al., 2000) demandant que tous les nœuds du réseau participent à la surveillance. Cette solution donne de bons résultats pour la détection de nœuds malveillants mais n'utilise pas les ressources de façon optimale.

3) Organisation du mémoire

Ce mémoire est organisé comme suit : le chapitre 1 présente les réseaux Ad-Hoc ainsi que les différents types de protocole.

La revue de littérature est présentée dans le chapitre 2 qui met en évidence d'une part les différents types d'attaques contre le protocole OLSR et d'autre part les différentes solutions proposées pour contrer ces attaques.

Le chapitre 3 est consacré au protocole SC-OLSR qui est une version modifiée de OLSR. La seconde partie du chapitre 3 est consacrée au modèle d'attaque de SC-OLSR et nous proposons des mécanismes de détection d'intrusion basés sur des critères relatifs aux spécifications du protocole SC-OLSR.

Finalement, nous avons réalisé des différentes simulations sous ns2 (*Network Simulator*) pour comparer l'efficacité du système de détection d'intrusion proposé dans ce projet de recherche avec celui proposé dans le mécanisme *Watchdog* (Marti et al., 2000). Nous comparerons aussi la stratégie de déploiement des nœuds dits moniteurs (nœud de surveillance) par rapport à celle présentée dans l'approche *Watchdog*.

En conclusion, nous présenterons les travaux réalisés ainsi que les résultats obtenus. Les recommandations pour les travaux futurs seront aussi présentées.

4) Diffusion des résultats

Les résultats de ce mémoire ont été présentés en octobre 2011 à la 7th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WIMOB).

CHAPITRE 1

LES RESEAUX AD-HOC

1.1 Historique

L'histoire des réseaux Ad-Hoc a commencé en 1972 avec une initiative du département de la défense américaine qui a subventionné le projet PRNET (*Packet Radio Network*) (Kahn et al., 1978). Par la suite, ce projet a donné naissance au projet SURAN (*Survivable Adaptative Radio Network*) (Beyer, 1990) au début des années 90. Le but de ces deux projets était de créer un réseau sans fil et sans infrastructure capable d'acheminer les données par voie radio souvent utilisée dans le domaine militaire.

L'introduction du standard 802.11 par l'IEEE a ouvert les portes pour une utilisation des réseaux Ad-hoc dans les applications civiles.

Le groupe MANET (*Mobile Ad-Hoc Network*) a été fondé au sein de l'IETF. Ce groupe a pour but de standardiser tout protocole de routage Ad-Hoc. Grâce au travail du groupe MANET, plusieurs applications ont vu le jour tant dans le domaine militaire que civile.

1.2 Applications des réseaux Ad-Hoc

La nature adaptative et autonome de la technologie Ad-Hoc a permis son utilisation lors de situations de conflit, des situations de désastre ou encore dans des applications pour le domaine de transport ou minier.

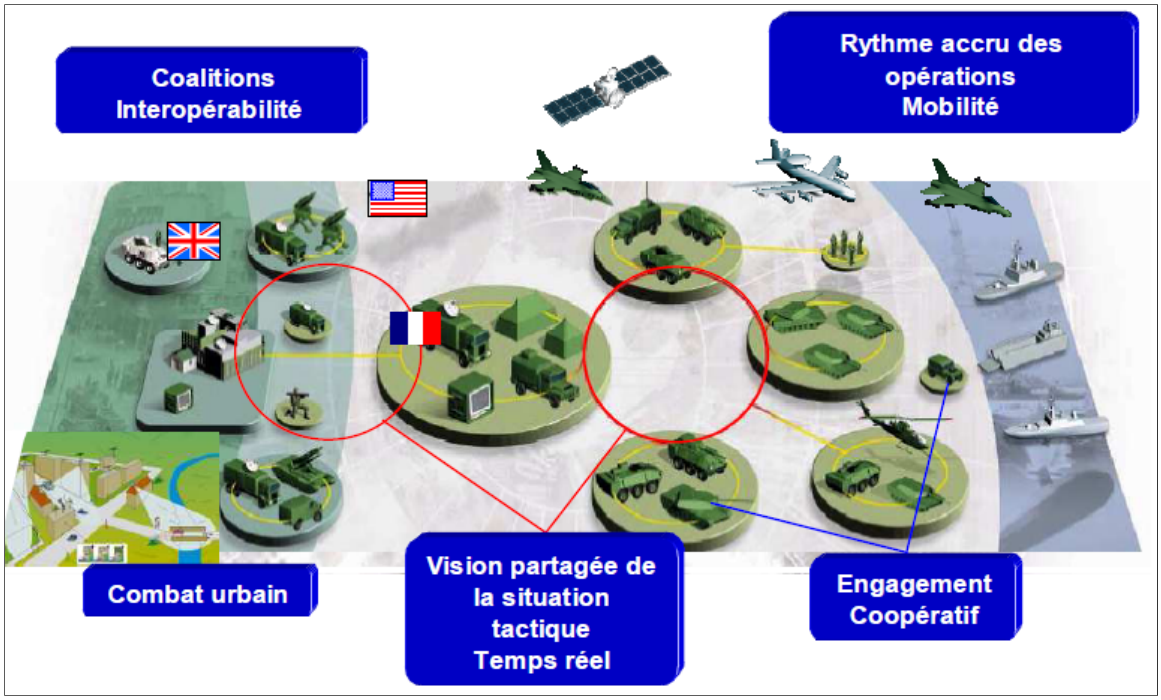

Figure 1.1 Application des réseaux Ad-Hoc dans le domaine militaire
Tirée de Herbiet (2007, p. 20)

La figure 1.1 montre les possibilités d'interaction entre les différentes unités (soldat, char et avion) que pourrait offrir une structure de communication basée sur la technologie Ad-Hoc.

Les réseaux Ad-Hoc peuvent aussi être utilisés dans les situations de désastre. Dans ce genre de situation, un réseau de communication doit être mis en place de façon rapide avec un minimum de configuration permettant de communiquer entre les unités de sauvetage. Cette technologie répond à ce besoin puisqu'elle ne nécessite pas d'infrastructure et offre une flexibilité et des possibilités d'auto-configuration.

La technologie VANET (*Vehicular Ad-hoc NETwork*) (Hartenstein et Laberteaux, 2008) a été proposée comme une variante de réseaux MANET appliquée au domaine du transport. Cette application est considérée comme l'une des applications les plus prometteuses de la technologie Ad-Hoc. Elle pourrait révolutionner la communication inter-véhiculaire avec des

systèmes intelligents (régulation de trafic, prévention d'accident,...) et aussi avec l'avènement des ordinateurs embarqués à l'intérieur des véhicules.

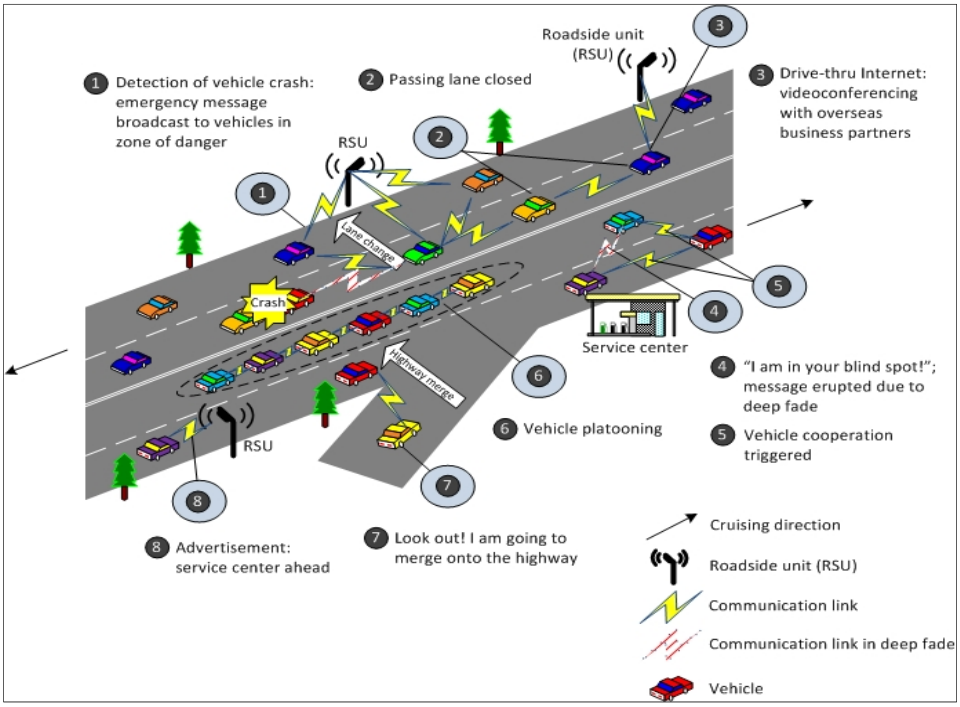

Figure 1.2 Application des réseaux Ad-Hoc dans le domaine du transport
Tirée de Cheng et al. (2010)

La figure 1.2 montre l'utilisation des MANET dans le domaine du transport. Les VANET seront sans doute une brique essentielle dans les projets ITS (*Intelligent Transport System*) (Cheng et al., 2010) visant à créer des applications de communication pour gérer dynamiquement le trafic et la signalisation routière.

WSN (*Wireless Sensor Network*) (Yick, Mukherjee et Ghosal, 2008) est une autre application des réseaux Ad-Hoc dans le secteur minier/industriel. Chaque nœud d'un réseau WSN est un capteur de données physiques (la température, la vitesse, la pression,…). Cet ensemble de capteurs dits intelligents pourrait échanger et analyser les données transmises/reçues et voire même communiquer avec l'entité de prise de décision. Ceci pourrait ajouter plus

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

d'autonomie et moins d'interventions humaines dans un secteur périlleux comme le secteur minier.

1.3 Caractéristiques des réseaux Ad-Hoc

Toutes les caractéristiques d'un réseau de communication sans fil sont applicables aux réseaux Ad-Hoc, quoique, il existe certaines propriétés propres à ce type de réseau à savoir :

- **mobilité** : la mobilité est une caractéristique essentielle des réseaux Ad-Hoc suivant laquelle les performances des protocoles de routage pourraient varier. La mobilité des nœuds rend les réseaux Ad-Hoc très versatiles surtout dans l'absence d'une entité centrale pour l'administration et la surveillance.

- **multi sauts** (*Multi Hoppin*g) : la propriété grâce à laquelle chaque nœud du réseau peut atteindre les autres nœuds hors portée radio grâce au protocole de routage.

- **auto-configuration** : l'absence d'une entité centrale d'administration exige que les nœuds doivent s'auto-configurer et s'auto-organiser afin de garantir la flexibilité et l'adaptabilité requises.

- **énergie** : la plupart des nœuds Ad-Hoc (ordinateur portable, PDA et capteurs) sont limités en matière d'énergie ce qui affecte la durée de vie de ces nœuds surtout si on considère la nature collaborative des protocoles de routage Ad-Hoc.

- **qualité de service** (QoS) : les applications gourmandes en ressources et surtout celles qui exigent une exécution en temps réel (VoIP, IPTV, jeux en ligne) représentent un vrai défi pour les réseaux Ad-Hoc. Le caractère versatile des nœuds et les ressources d'énergie limitées pourraient nuire à la qualité de service (QoS) offerte à travers un réseau Ad-Hoc.

- **sécurité** : la sécurité dans l'environnement sans fil s'avère un grand défi en particulier pour les réseaux Ad-Hoc. L'information transite d'un nœud à un autre ce qui met en question la confidentialité, l'intégrité et la disponibilité des données dans un environnement où seule la confiance régit la transmission de données.

Nous allons présenter dans la section qui suit les différentes familles de protocoles de routage pour les réseaux Ad-Hoc avec leurs caractéristiques, avantages et faiblesses.

1.4 Protocoles de routage pour les réseaux Ad-Hoc

Compte tenu des caractéristiques des réseaux Ad-Hoc présentées dans la section précédente, la conception d'un protocole de routage Ad-Hoc s'avère un grand défi consistant à trouver un compromis entre les performances souhaitées et les limitations des réseaux Ad-Hoc. Il existe deux grandes familles de protocoles de routage Ad-Hoc à savoir les protocoles proactifs et réactifs. Il existe une famille hybride constituée d'un mélange de ces deux protocoles. Cette famille est appelé famille des protocoles hybrides.

1.4.1 Protocoles proactifs

Les protocoles proactifs maintiennent une table de routes entre chaque paire de nœuds. Le but de cette stratégie est de fournir instantanément une route déjà stockée entre la source et la destination aussitôt que le besoin se présente. La famille de protocoles proactifs se divise en deux catégories. La technique utilisée pour la découverte et le maintien des liens différencie ces deux catégories. La première catégorie est appelée protocoles avec ***vecteur distance*** où la métrique pour le calcul des routes est le nombre de sauts séparant la source à la destination. La deuxième catégorie est appelée protocoles à ***état de lien*** utilisant l'état des liens pour le calcule de routes. La mise à jour des routes est faite par des techniques de diffusion (*Broadcast*). Les protocoles DSDV (protocole avec vecteur distance) et OLSR (protocole avec état de lien) sont des exemples de ces deux catégories.

Rapport-Gratuit.com

DSDV (*Destination-Sequenced Distance Vector*) présenté par Perkins et Bhagwat (1994) est l'un des premiers protocoles destinés aux réseaux Ad-Hoc. Inspiré du protocole RIP (*Routing Information Protocole*), le protocole DSDV a réglé le problème de convergence de route (boucle de routage infinie) dont RIP souffrait (Perkins et Bhagwat, 1994) en introduisant un numéro de séquence dans chaque entrée de la table de routage. Ce numéro est géré par le nœud destination. La mise à jour des routes se fait en utilisant un numéro de séquence pour assurer l'unicité des messages de mise à jour. Les routes sont enfin calculées par l'algorithme de Bellman-Ford.

OLSR (*Optimised Link State Routing*) proposé par Clausen et Jacquet et standardisé dans la RFC 3626 (Clausen et Jaquet, 2003), est une version optimisée du protocole LSR (*Link State Routing*) (McQuillan, Richer et Rosen, 1980). La majeure contribution consiste à introduire le concept de nœuds relais MPR (*Multipoint Relay*). Avec ce concept, la diffusion des messages de données ou de routage (ajout/suppression de lien, mise à jour des tables,...) est optimisée. Un MPR joue le rôle de représentant pour joindre les nœuds qui l'ont choisi (la transmission par inondation est donc évitée). Ainsi avec ce protocole, les ressources sont largement optimisées. À noter que les routes sont calculées en utilisant l'algorithme Djikstra. Ce protocole sera détaillé dans une autre section de ce chapitre.

1.4.2 Protocoles réactifs

À l'opposé des protocoles proactifs, les protocoles réactifs adoptent une approche différente. En effet, ils proposent de déclencher la phase de découverte de routes au besoin (lors de l'envoi d'un message) d'où l'appellation *On Demand Protocol*. L'avantage de cette approche consiste à éliminer le coût de maintenance des tables de routage ainsi que la surcharge du réseau (pas de mise à jour de route requise). Toutefois, cette approche souffre de problèmes de délais de transmission de paquets engendrés par le processus de découverte de route lui-même. L'envoi des messages est retardé jusqu'au moment de trouver une route vers la destination. Le protocole DSR est un exemple de ce type de protocole.

DSR (*Dynamic Source Routing*) (RFC 4727) par Johnson, Maltz et Broc (2001) est un protocole où le processus de découverte de route est initié par la source. La source envoie un message de demande de route de type *Route Request* à l'ensemble du réseau. Ces messages sont rediffusés par les nœuds internes jusqu'à atteindre la destination ou un nœud qui connait un chemin vers cette destination. La réponse se fait avec un message *Route Reply* vers le nœud source qui, une fois qu'il a reçu les différents messages *Route Reply*, compile et sélectionne la route optimale.

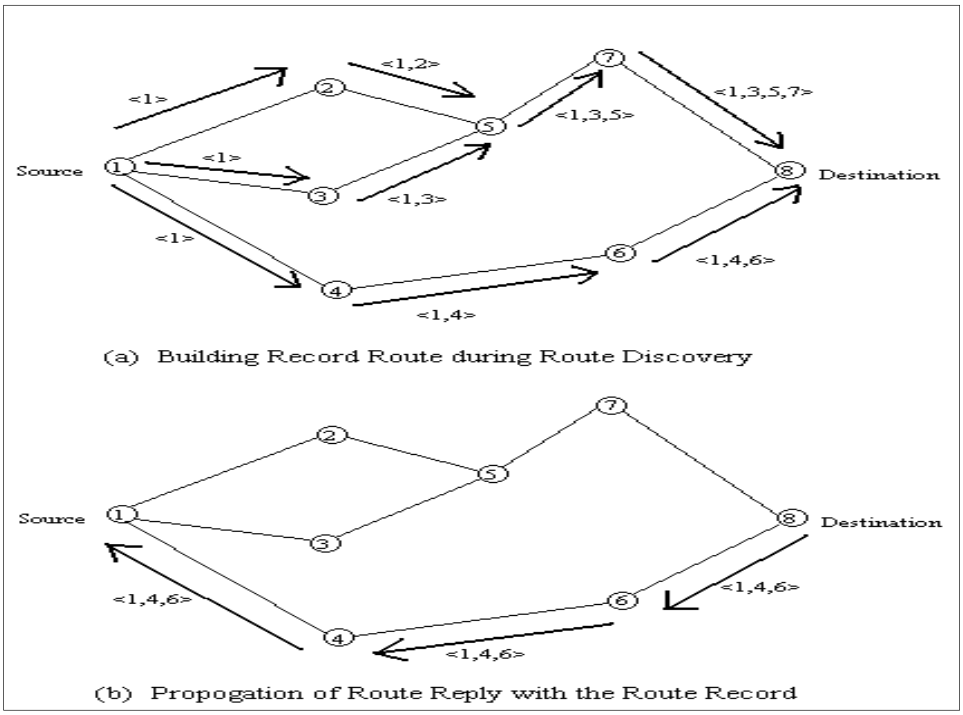

Figure 1.3: Phase de découverte de routes dans le protocole DSR
Tirée de Johnson et al. (2001)

1.4.3 Protocoles hybrides

Pour ce type de protocole, une combinaison de l'approche des protocoles réactifs et proactifs est adoptée afin de tirer les avantages de chaque approche. Les nœuds se comportent de façon proactive localement dans une zone déterminée par le nombre de sauts maximum et de façon

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

réactive à l'extérieur de cette zone. Le protocole ZRP est un exemple de cette famille de protocole.

ZRP (*Zone Routing Protocol*) (Beijar, 2002) est parmi les premiers protocoles hybrides proposés avec des composantes des protocoles proactifs et réactifs à la fois. ZRP a été proposé pour réduire la surcharge causée par les protocoles proactifs et réduire aussi les délais existants dans les protocoles réactifs. Le protocole divise le réseau en deux zones déterminées par un nombre de sauts X. Un protocole de routage proactif est utilisé à l'intérieur de la zone nommée IARP (*Intrazone Routing Protocol*) et un autre réactif à l'extérieur de cette zone notée IERP (*Interzone Routing Protocol*).

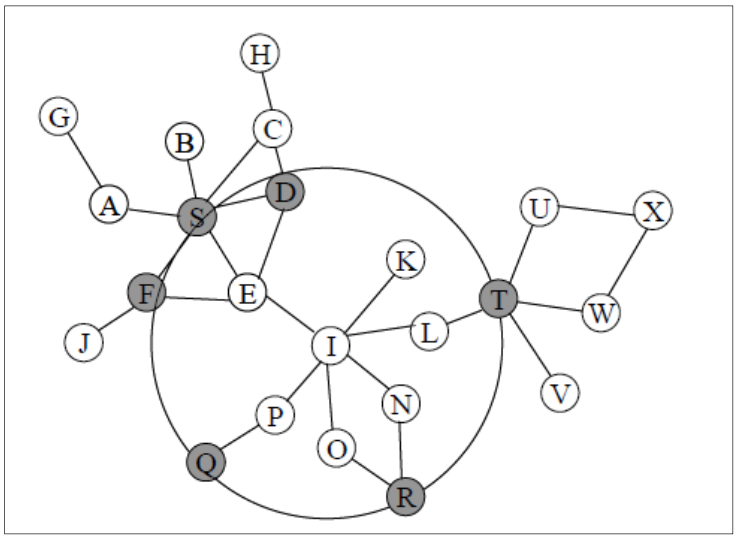

Figure 1.4 Fonctionnement du protocole ZRP
Tirée de Beijar (2002)

1.5 Description détaillée des protocoles OLSR et SC-OLSR

1.5.1 Protocoles OLSR

Le protocole OLSR (*Optimised Link State Routing*) fait partie de la famille des protocoles proactifs développée par l'équipe Hipercom de l'INRIA. Ce projet a été standardisé par

l'IETF en 2003 (Clausen et Jansen, 2003). Le protocole est à sa deuxième version. Des modifications ont été apportées afin de supporter de nouvelles fonctionnalités.

Le protocole OLSR a introduit une nouvelle technique de diffusion des messages pour les réseaux Ad-Hoc. Cette technique consiste à optimiser le processus d'inondation (*Flooding*) nécessaire pour un protocole de routage en introduisant la notion des nœuds relais MPR. Les MPR sont des nœuds spéciaux sélectionnés par leurs voisins pour relayer les messages de service et données à travers tout le réseau.

Le fonctionnement du protocole OLSR est divisé en trois parties : la découverte des nœuds du voisinage, la découverte et la mise à jour de la topologie du réseau et, enfin, le calcul des routes.

1) Découverte des nœuds de voisinage

Comme presque tous les protocoles de routage pour les MANET, le protocole OLSR propose une phase de découverte des nœuds de voisinage consistant à envoyer à ses voisins directs à un saut (voisins qui sont dans la portée radio) la liste de tous ses voisins directs. Cette liste contient non seulement les adresses des voisins, mais aussi le type de lien et le type de relation de voisinage. Ces informations sont encapsulées dans un message spécial appelé message HELLO. La figure 1.5 montre les différents champs constituant ce message. Les principaux champs sont :

- *reserved* : champ réservé pour utilisation future.

- *htime* : intervalle d'émission des messages HELLO.

- *willingness* : ce champ traduit l'habilité d'un nœud à relayer un message vers un autre nœud. cette valeur est utilisée pour le processus de sélection des nœuds MPR.

- *link code* : code identifiant le type de lien entre l'expéditeur et les interfaces listées des nœuds voisins. Ce champ a une taille de 8 bits, mais seuls les 4 premiers bits sont utilisés pour designer le type de relation de voisinage et le type de lien.

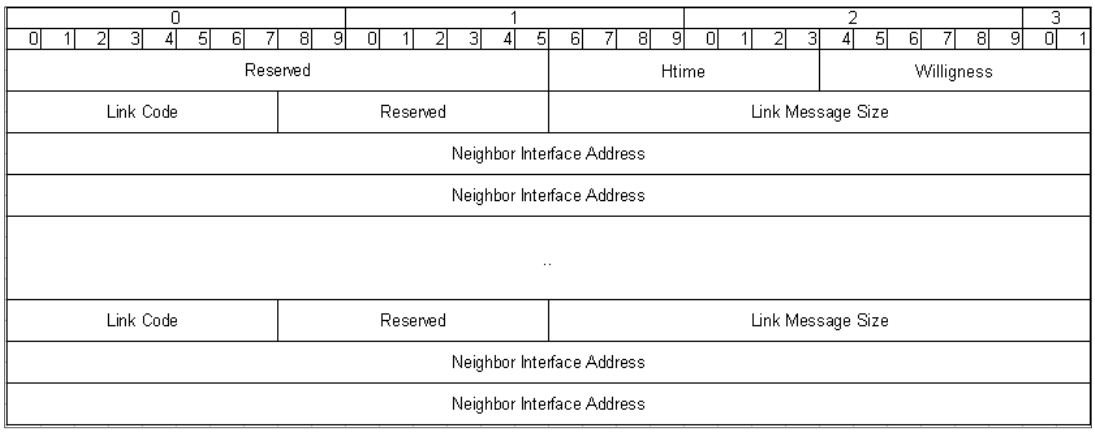

Figure 1.5 Champs du message HELLO
Tirée de RFC 3626

À la fin de cette phase, chaque nœud connait non seulement l'ensemble de ses voisins directs à un saut noté N1H, mais aussi l'ensemble de ses voisins à deux sauts noté N2H (les voisins de ses voisins) en analysant les messages HELLO échangés. Pour chaque nœud existant dans la liste N1H ou N2H, il est possible de connaitre le type de lien et aussi la nature du type de voisin (symétrique ou pas un MPR) grâce au champ *Link Code.*

Tableau 1.1 Valeurs possibles de champ *Link Code* selon RFC 3626

Type de lien	
UNSPEC_LINK	Pas d'information
ASYM_LINK	Lien asymétrique
SYM_LINK	Lien symétrique
LOST_LINK	Lien perdu

Type de voisin	
SYM_NEIGH	Voisin symétrique
MPR_NEIGH	Voisin a été sélectionné comme MPR
NOT_NEIGH	Pas de voisins/ Pas encore symétrique

Ces informations sont stockées dans des structures de données à partir desquelles les nœuds MPR sont sélectionnés en utilisant un algorithme vorace. À noter que les messages HELLO ne sont destinés qu'aux nœuds voisins (à un saut) de l'expéditeur. Ils ne doivent donc jamais être rediffusés par un MPR d'où la valeur du champ *Time To Live* (TTL) égale à 1.

La figure 1.6 montre comment la phase de découverte des voisins est faite dans le protocole OLSR. Après avoir découvert les voisins directes N1H A et C, le nœud B envoie un message HELLO au nœud A contenant sa liste de ses voisins N1H (A et C). Le nœud A quand à lui analyse le message HELLO et ajoute le nœud C à sa liste N2H (voisin de voisin). Le nœud C procède de la même façon.

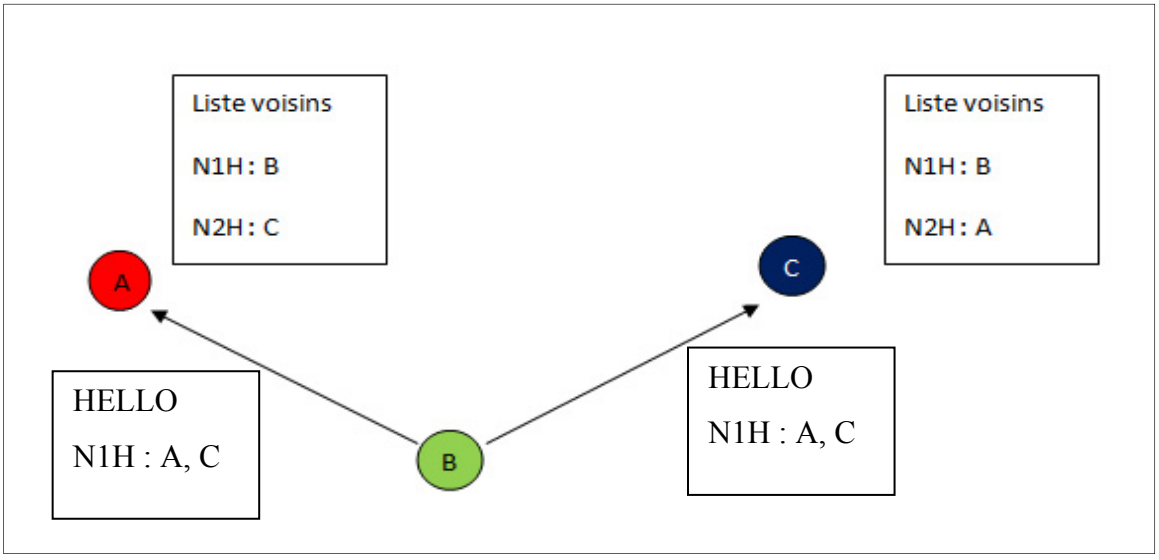

Figure 1.6 : Découverte de voisins N1H et N2H avec le message HELLO

2) Construction et contrôle de topologie

Une fois que les nœuds MPR sont sélectionnés, ces derniers diffusent (*Broadcast*) un message au reste du réseau (TTL=255) afin de les informer de l'ensemble des nœuds qui les ont choisis comme MPR. Cet ensemble est appelé *MPR Selectors*.

Le message envoyé par les nœuds MPR est appelé message TC (*Topology Control*) qui regroupe la liste des *MPR Selectors*. Ce message est envoyé à tous les voisins du nœud MPR expéditeur. Toutefois, seuls les nœuds MPR ont le droit de relayer ce message TC. En procédant ainsi, le processus d'inondation du réseau est optimisé comme le montre la figure 1.7. Dans cet exemple, seulement 8 nœuds sur 16 relaient les messages du nœud central.

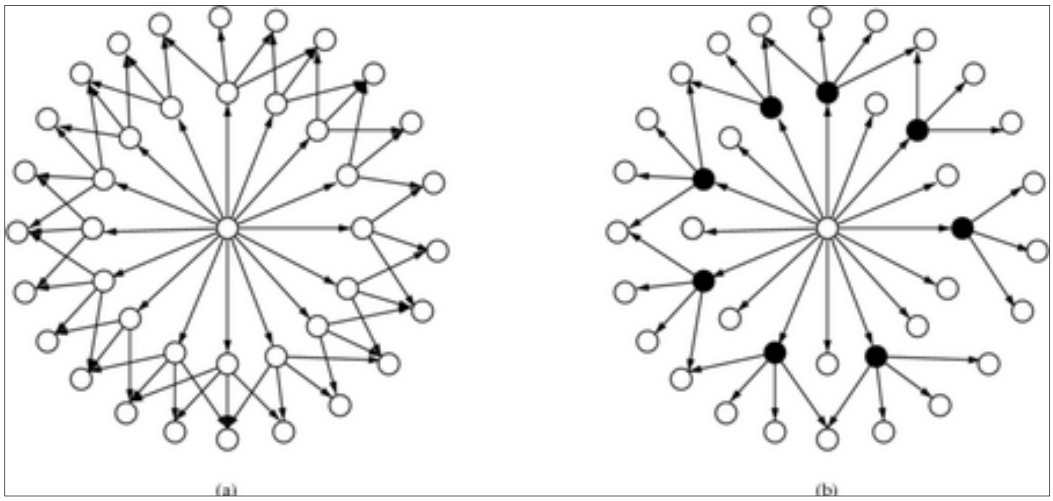

Figure 1.7 (a) Processus d'inondation classique**,** (b) Processus d'inondation MPR
Tirée de Clausen et al. (2003)

On retrouve dans la figure 1.8 les différents champs d'un message TC :

- *reserved* : pour les utilisations futures.

- *ANSN* (*Advertised Neighbor Sequence Number*) : code généré par la source à partir duquel les routes sont mises à jour par la destination.

- *advertised Neighbor Main Adress* : champ contenant l'adresse d'un MPR *Selector*.

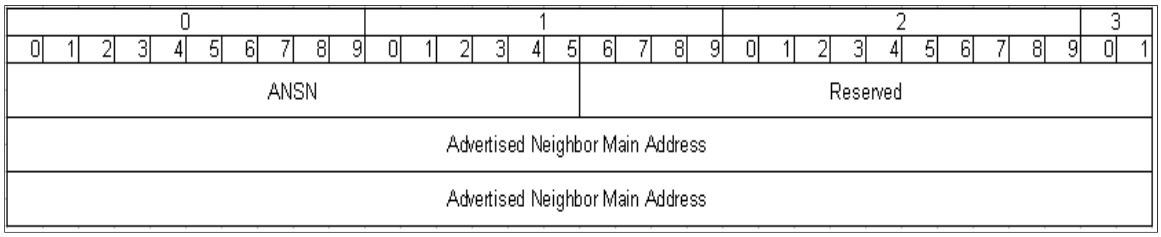

Figure 1.8 Champs du message TC
Tirée de RFC 3626

Grâce au concept des nœuds MPR et des messages TC, il est possible de transiter de la topologie locale obtenue lors de la phase de découverte des nœuds par transmission de messages HELLO, à une topologie globale où chaque nœud peut connaitre la topologie du graphe sous-jacent connectant tous les nœuds du réseau.

3) Calcul des routes

En fonction de la densité du réseau, plusieurs routes pourraient être choisies afin d'atteindre une destination. L'algorithme Djikstra est utilisé pour déterminer le choix d'une route optimale (le plus court chemin) basée sur l'état des liens. Le calcul des routes est répété à chaque fois qu'un changement de topologie est remarqué dû à une perte de lien avec un voisin (message HELLO) ou à un changement dans la liste des MPR *Selectors.* Ceci permet la mise à jour automatique des tables de routage.

Tableau 1.2 Résumé des phases de protocole OLSR

Phase	Message	Exécutée par	Résultat
Découverte de voisin	HELLO	Tous les nœuds	Connaissance des nœuds N1H et N2H et sélection de MPR.
Construction et contrôle de topologie globale	TC	MPR	Connaissance de topologie globale
Calcul des routes	____	Tous les nœuds	Calcule de tables de routage

1.5.2 Le protocole SC-OLSR

Le protocole OLSR est certes l'un des protocoles les plus utilisés dans la technologie Ad-Hoc dans applications civiles ou militaire. Toutefois, ce protocole souffre de certaines faiblesses à savoir :

- le nombre élevé de nœuds MPR choisis.
- la surcharge du réseau par les messages envoyés par les MPR due au nombre élevé de nœuds MPR.
- les nœuds MPR sont choisis en fonction de leur connectivité et non de leur niveau d'énergie. Ainsi, les mêmes nœuds peuvent être choisis fréquemment.
- les nœuds peuvent avoir tendance à limiter leur consommation d'énergie en refusant de participer aux différents processus collaboratifs (retransmission de trafic, rôle de MPR) dans le but de préserver leurs ressources.

Ces divers problèmes ont pour conséquence de dégrader les performances du réseau et réduire la durée de vie des nœuds ayant un faible niveau d'énergie.

Les faiblesses du protocole OLSR ont poussé les chercheurs à proposer des modifications aux spécifications de ce protocole. Une des approches proposées est de regrouper les nœuds en grappe. Cette approche consiste à élire un nœud spécial répondant à certains critères pour représenter un ensemble de nœuds et se charger de certaines tâches (sélection de MPR, émission de TC, retransmission de trafic). Ce nœud est appelé chef de grappe ou *Cluster Head* (CH). Ce concept vise à réduire le nombre de messages TC émis en réduisant le nombre de nœuds MPR. Ceci est possible en votant pour un CH en fonction de certaines métriques. En considérant la lourdeur des tâches à effectuer par un CH, les nœuds ayant suffisamment d'énergie ont tendance à ne pas dévoiler leurs ressources affectant ainsi le bon fonctionnement de ce type de protocole.

SC-OLSR (Chriqi, Otrok et Robert, 2009) est l'un des protocoles adoptant l'approche de grappe où les CH sont choisis localement par chaque nœud. Les critères d'élection d'un CH sont l'index de connectivité (*connectivity index*) et l'énergie résiduelle. Les résultats montrent que SC-OLSR arrive à prolonger davantage la durée de vie du réseau comparé à HOLSR (*Hierarchical* OLSR). SC-OLSR utilise des moyens d'incitation pour encourager les nœuds à déclarer honnêtement leurs ressources et surtout à participer aux différents processus collaboratifs. Les moyens d'incitation utilisés ont été inspirés des mécanismes VCG (*Vickrey-Clarke-Groves)* utilisés pour les enchères. Les nœuds participants aux processus reçoivent une récompense qui pourrait être utilisé pour la transmission de leurs propres paquets.

La figure 1.9 présente le mécanisme de construction des grappes dans SC-OLSR. Chaque nœud vote pour le nœud présentant les ressources les plus élevées parmi les voisins à un saut.

Le modèle utilisé pour représenter les ressources est un modèle hybride qui réunit l'énergie résiduelle et l'indice de connectivité. Ces deux paramètres sont réunis dans un seul paramètre

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

appelé CF ou CTF (*Cost to Forward*) ou le coût de transfert d'un paquet. Le modèle utilisé est présenté dans la figure 1.10.

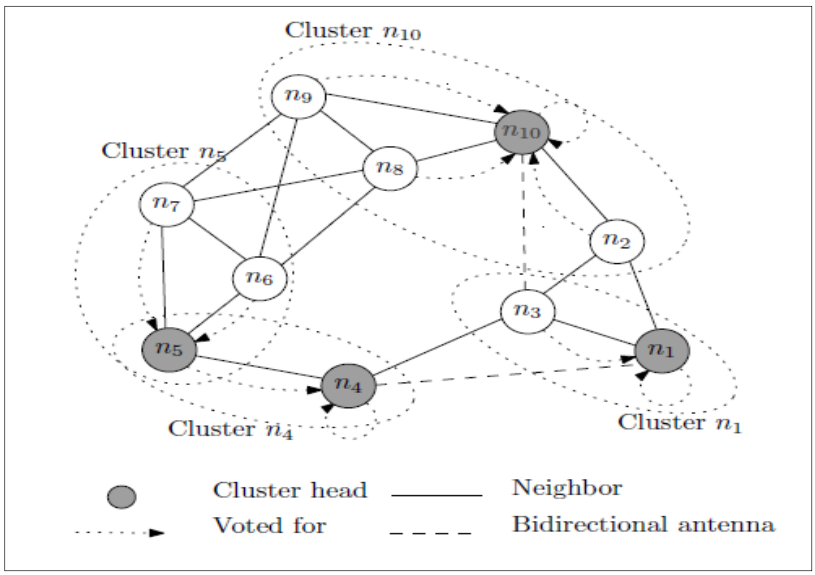

Figure 1.9 Principe de vote pour un CH dans SC-OLSR
Tirée de Chriqi et al. (2009)

Notation and Cost-to-Forward Function

Let i be a node in the network. Let define

$CFP(i)$	=	Its cost to forward one packet
$N_1(i)$	=	Its 1-hop neighbors
$N_2(i)$	=	Its 2-hop neighbors
$N_{1,2}(i)$	=	$N_1(i) \cup N_2(i)$
$RE(i)$	=	Its residual energy

1 *Connectivity Model*

$$CF(i) = CFP(i) \times \frac{1}{|N_1(i)|}$$

2 *Energy Model*

$$CF(i) = CFP(i) \times \frac{1}{RE(i)}$$

3 SC-OLSR *Hybrid Model*

$$CF(i) = CFP(i) \times \frac{1}{|N_1(i)| \times RE(i)}$$

Figure 1.10 Modèle hybride de représentation du cout de transfert.
Tirée de Chriqi et al. (2009)

Format des messages

Pour supporter les différentes fonctionnalités présentées dans SC-OLSR, des modifications ont été effectuées aux anciens messages du protocole OLSR (messages HELLO et TC), un nouveau message a été créé pour implanter le concept de grappes. Ce message sert à annoncer à tous les voisins N1H, le nœud choisi comme CH. Ce nouveau message est appelé le message ELECTION.

1) Message HELLO

La figure 1.11 présente les différentes modifications apportées au message HELLO. Les nouveaux champs ajoutés sont :

- *reserved* : un champ de 8 bits, les trois derniers bits servent à designer la version du protocole et le premier bit noté H est un fanion indiquant si le message HELLO a été envoyé par un CH.

- *willingness* : ce champ est utilisé pour envoyer la valeur du coût de l'envoi (CF) d'un nœud suivant le modèle hybride proposé de SC-OLSR.

- *link code* : dans le protocole OLSR classique, ce champ est divisé en deux parties: le type de lien et le type de voisin. Dans SC-OLSR, une autre valeur pour le type de voisin a été introduite pour mentionner que le voisin en question a été élu comme CH.

- *payment* : cette valeur représente le moyen d'incitation (paiement) pour le nœud participant aux actions collaboratives (retransmission de paquet, CH, MPR).

À noter que dans chaque message HELLO, on trouve l'adresse de tous les voisins distants d'un saut avec l'adresse de leur CH. La valeur ajoutée du champ *Link Code* indique si le voisin a été élu comme CH avec son coût de transfert de paquet (*Cost to forward*).

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

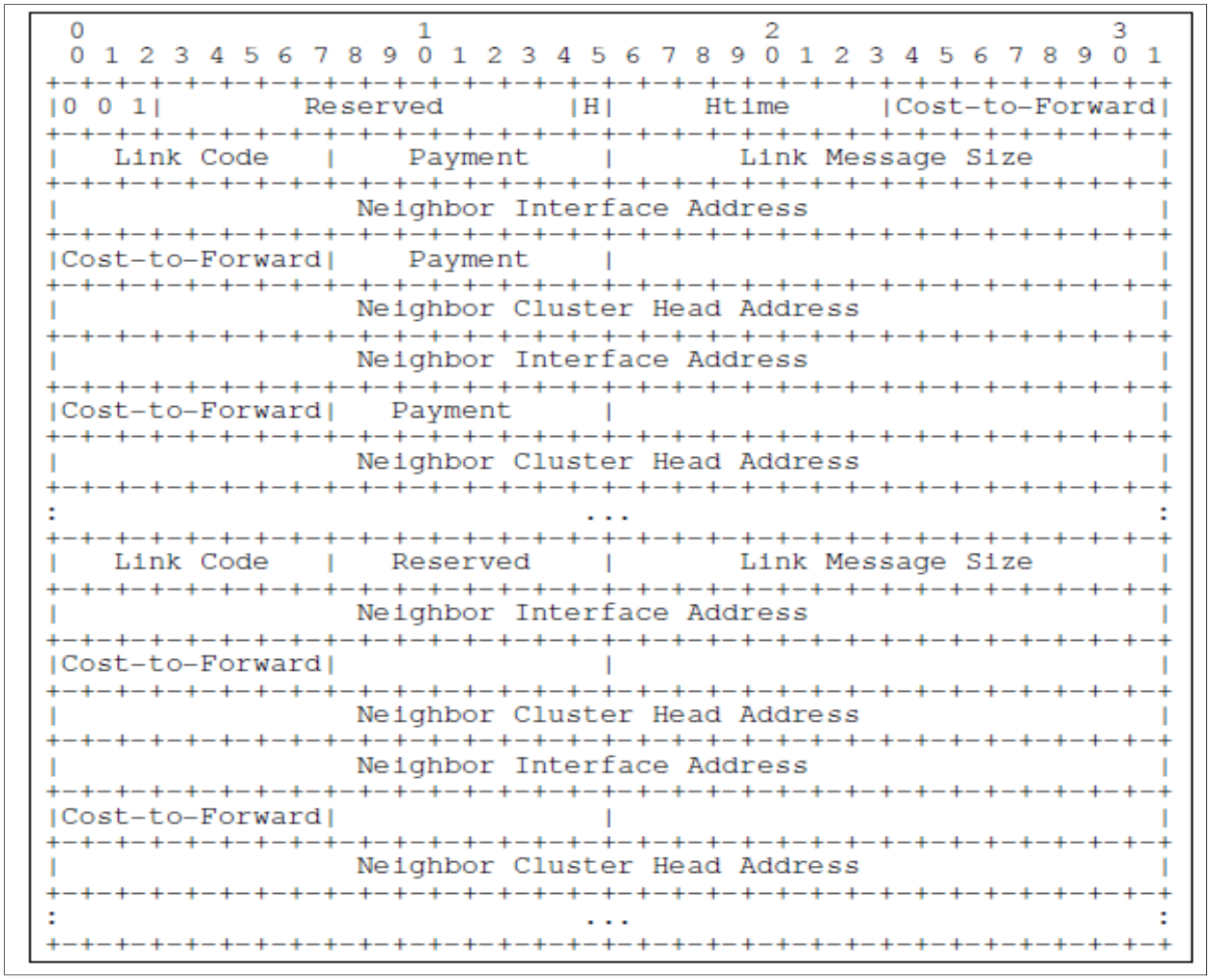

Figure 1.11 Message HELLO modifié
Tirée de Chriqi et al. (2009)

2) Message TC

Le message TC a été modifié en introduisant deux champs supplémentaires nommés REPUTATION et COST-to-FORWARD. Ces champs sont utilisés pour choisir un chemin en fonction de la réputation du nœud (une valeur qui varie en fonction du comportement malicieux/honnête du nœud) et des ressources requises pour retransmettre un paquet (voir figure 1.12).

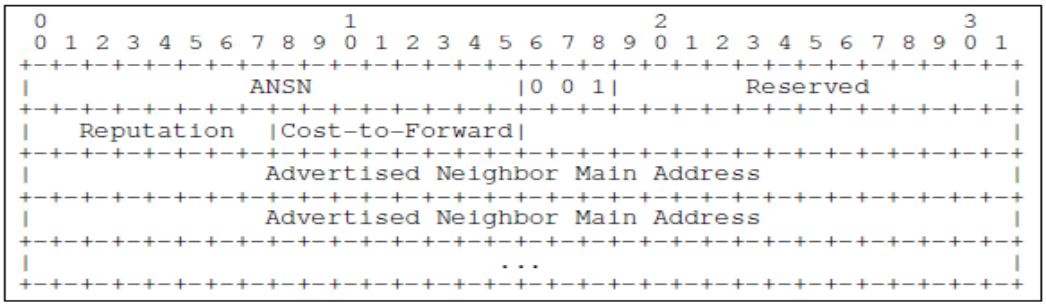

Figure 1.12 Message TC modifié
Tirée de Chriqi et al. (2009)

3) Message d'élection

Comme expliqué précédemment, le concept de grappe consiste à élire un représentant de grappe appelé CH. Pour ce faire, un message ELECTION a été ajouté. Ce nouveau type de message sert à annoncer à tous les voisins existant à un saut l'adresse du CH choisi. La figure 1.13 montre les différents champs composant ce nouveau message. Un champ *payment* a été ajouté à ce message. Il représente la récompense offerte à ce nœud pour avoir accepté de servir comme CH pour le nœud émetteur du message ELECTION. Cette récompense peut être utilisée par ce nœud pour transmettre ces propres paquets. D'où l'importance du mécanisme d'incitation utilisé dans les différents processus collaboratifs du réseau.

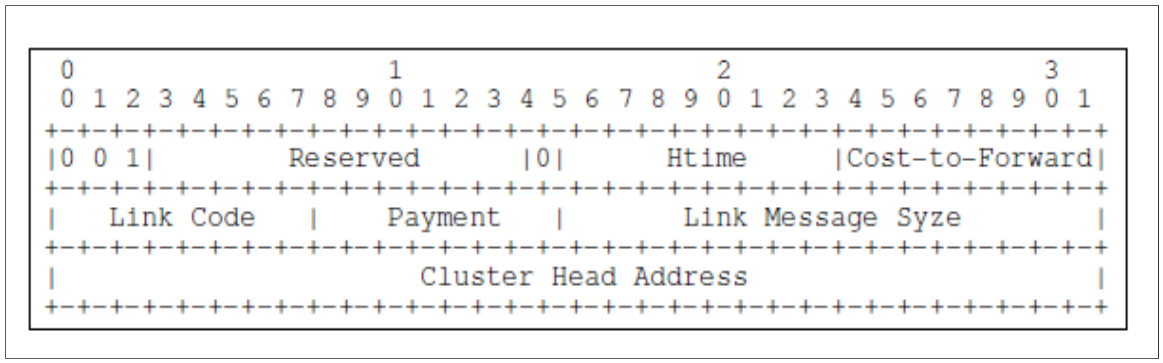

Figure 1.13 Le nouveau message ELECTION
Tirée de Chriqi et al. (2009)

CHAPITRE 2

ATTAQUES ET VULNÉRABILITÉS CONTRE LES PROTOCOLES OLSR/SC-OLSR

2.1 Sécurité dans les réseaux Ad-Hoc : généralités

Après avoir présenté dans le chapitre 1 les caractéristiques et les particularités des réseaux Ad-Hoc, nous présenterons à travers ce chapitre un des points faibles des réseaux Ad-Hoc : la sécurité.

Nous essayerons de caractériser les différents aspects de la sécurité dans le protocole SC-OLSR en étudiant les vulnérabilités de ce protocole afin de quantifier leurs impacts.

Avant de concevoir tout mécanisme de sécurité, la phase préliminaire consiste à caractériser le système à protéger. Caractériser un système consiste à connaitre les vulnérabilités, les menaces et les attaques possibles contre ce système afin de préserver les trois objectifs de la sécurité : la confidentialité, l'intégrité et la disponibilité.

- vulnérabilité : toute lacune ou faille de nature matérielle ou logicielle qui pourrait être exploitée pour réaliser une attaque.

- menace : c'est la possibilité qu'une vulnérabilité soit exploitée accidentellement ou malicieusement par un agent.

En général, les attaques sont classées en deux catégories :

- attaque externe : une attaque exercée par un nœud qui ne possède pas l'autorisation requise pour accéder au réseau. De ce fait, un attaquant externe est incapable d'envoyer des messages authentifiés mais il est capable de lancer des attaques de rejeux, attaques

consistant à capturer et renvoyer les messages des autres nœuds (medium ouvert) pour contourner les phases d'authentification.

- attaque interne : une attaque exercée par un nœud compromis ayant le droit d'accès au réseau. Ayant les clés d'authentification nécessaires, ce type de nœuds peut exercer des attaques plus élaborées que les attaquants externes (redirection de trafic, suppression, altération,…).

La sécurité dans un réseau Ad-Hoc diffère beaucoup de celle des réseaux filaires. En effet, la nature ouverte du lien physique utilisé complique davantage la conception d'un système fiable du point de vue de la sécurité. Les paquets sont échangés dans les airs. Par conséquent, tout nœud existant dans le rayon de la portée radio peut connaitre au moins les attributs physiques de la communication (puissance d'émission, modulation, codage,…), permettant ainsi de décoder le paquet et connaitre son contenu sans que la source et la destination se rendent compte. De telles interceptions ne sont possibles que si le medium physique (câble, fibre optique) est brisé dans les réseaux filaires.

La nature versatile des réseaux Ad-Hoc présente un grand défi pour protéger de tels systèmes. L'origine de ce problème est due à l'absence de spécification de sécurité lors de la définition des protocoles de routage pour les réseaux Ad-Hoc. De plus, l'absence d'un élément central statique dans les réseaux Ad-Hoc qui, traditionnellement, analyse tous les paquets transitant par lui et qui joue souvent le rôle de détecteur d'intrusion (IDS) dans les réseaux filaires rend la protection du réseau Ad-Hoc très complexe.

Les vulnérabilités des réseaux Ad-Hoc ne se limitent pas malheureusement au problème de lien physique ouvert mais aussi aux mécanismes de routage et d'auto-configuration utilisés. Ces mécanismes sont basés sur la confiance entre les nœuds participants. S'il s'avère qu'un nœud présente un comportement malicieux, tous les services coopératifs offerts par le réseau seront paralysés (empoisonnement de table de routage, congestion, altération de paquets,…).

De ce qui précède, nous constatons que mettre en œuvre un système de sécurité pour les réseaux Ad-Hoc est une nécessité absolue. Ce système doit tenir compte des limitations de la technologie Ad-Hoc.

2.2 Vulnérabilités et types d'attaques contre le protocole OLSR

La RFC 3626 (Clausen et Jansen, 2003), document décrivant le protocole OLSR, n'a pas établi de requis pour la sécurité. Par conséquent, le protocole souffre de plusieurs vulnérabilités qui peuvent être exploitées pour attaquer et nuire au bon fonctionnement du système.

Selon Adjih et al. (2005a) les attaques contre le protocole OLSR se divisent en deux catégories : attaques contre le protocole de routage et les attaques de manipulation de trafic.

2.2.1 Attaques contre le protocole de routage

Selon Adjih et al. (2005a) les attaques contre les messages de service (messages TC et HELLO) visent à compromettre les entrées de la table de routage ou modifier la topologie du réseau (la liste des voisins ou la liste des nœuds MPR). Le but ultime de ces attaques est de rediriger les flots de trafic vers le nœud malicieux afin de les exploiter à des fins frauduleuses ou encore isoler des nœuds du réseau. Deux techniques sont souvent utilisées :

- **usurpation d'identité** (*Identity Spoofing*) : le nœud malveillant génère du trafic en prétendant être un autre nœud.

- **usurpation de lien** (*link spoofing)* : le nœud malveillant annonce avoir un faux lien de type voisin ou MPR avec un autre nœud en émettant des faux messages TC et HELLO.

Nous présenterons dans la partie qui suit les attaques contre les messages HELLO et TC avec leurs impacts sur le réseau.

2.2.1.1 Attaques contre les messages HELLO

- **génération de faux messages** : le processus de sélection de MPR passe par les messages HELLO en particulier le champ *Willingness*. Modifier la valeur de ce champ pourrait augmenter (attirer le trafic) ou réduire (comportement égoïste) la probabilité pour un nœud d'être sélectionné comme MPR en fonction de ses intentions malicieuses.

- **usurpation d'identité** : un nœud malicieux peut prétendre être un autre nœud en émettant des messages HELLO au nom de celui-ci. De ce fait, les liens de voisinage et les tables de routage seront modifiés (voir figure 2.1).

Un nœud malicieux peut usurper l'identité d'un nœud MPR en envoyant des messages HELLO avec des informations erronées au nom du nœud victime. Dans la figure 2.1, le nœud D prétend être le nœud C et envoie des messages HELLO aux nœuds E et F. Ces derniers croient qu'ils sont directement liés au nœud C.

- **usurpation de lien** : cette attaque peut être exécutée de deux façons différentes. Si le nœud malicieux prétend avoir des liens avec plusieurs voisins (des faux liens déclarés dans le message HELLO), il augmentera ses chances d'être élu comme MPR. S'il ne le déclare pas dans sa liste de voisinage dans le message HELLO, il pourra isoler un nœud dans le réseau (Abdellaoui et Robert, 2009).

2.2.1.2 Attaques contre le message TC

- **génération de faux messages**: si un nœud malicieux génère un faux message TC, les nœuds récepteurs de ce message le considèreront comme un MPR de tous les nœuds déclarés dans le MPR *Selector List.* Seuls les nœuds MPR ont le droit d'envoyer des messages TC créant des conflits dans la table de routage et modifiant la topologie par conséquence.

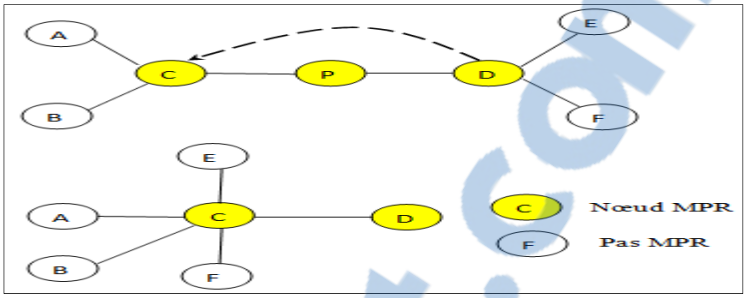

Figure 2.1 Usurpation d'identité avec le message HELLO

- **usurpation de lien** : un nœud malicieux sélectionné comme MPR peut lancer cette attaque de deux manières différentes. Soit ajouter des nœuds qui ne l'ont pas choisi comme MPR dans son MPR *Selector* List ce qui entrainera la création de faux liens. Soit de supprimer des nœuds de son MPR *Selector List* ce qui pourrait isoler le nœud supprimé du reste du réseau et modifier la topologie (voir figure 2.2).

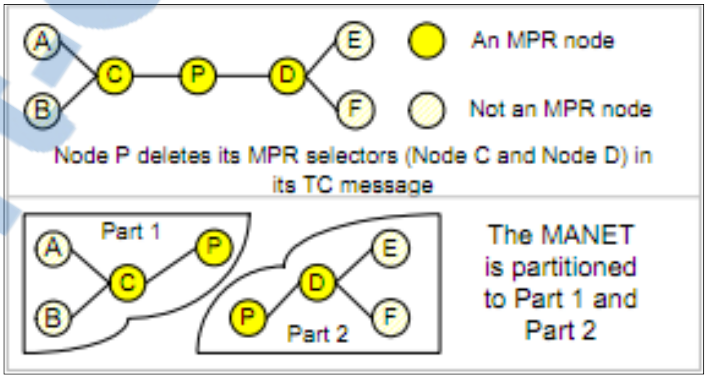

Figure 2.2 Usurpation de lien avec les messages TC
Tirée de Wang et al. (2005)

2.2.2 Attaques de manipulation de trafic de données

La manipulation ou la retransmission incorrecte du trafic de données est souvent classée comme la seconde étape d'une attaque complexe précédée par une modification de la topologie (voir la section précédente). Le nœud malicieux augmente la probabilité pour

Rapport-Gratuit.com

qu'un maximum de trafic passe par lui afin d'exploiter cette situation pour des fins frauduleuses.

- Écoute passive/ Altération/ Élimination

Le nœud malicieux dans ce cas analyse tout le trafic passant par lui à la recherche de données privées (clé de chiffrement mot de passe,...). Cette attaque est la conséquence directe de la propriété du lien physique ouvert (sans fil) utilisé dans le réseau Ad-Hoc.

2.2.3 Solutions proposées pour la sécurité du protocole OLSR

La sécurité des réseaux Ad-Hoc est un domaine de recherche en plein ébullition. Les chercheurs ont proposé plusieurs solutions pour assurer la sécurité de ce type de réseaux. Les solutions proposées varient entre des solutions cryptographiques pour contrer les attaques externes ou des solutions basées sur la surveillance pour détecter les attaques internes.

2.2.3.1 Cryptographie (attaques externes)

Système basé sur une infrastructure de distribution de clés publique et signature

Pour pallier les problèmes de messages altérés, une technique proposée par Adjih et al. (2003) pour assurer la sécurité de bout en bout consiste à ajouter des signatures dans les messages de protocole (messages HELLO et TC) qui seront vérifiées pour valider leur authenticité.

L'architecture proposée consiste à envoyer séparément le message signé et les messages HELLO et TC. L'authentification des messages HELLO et TC est vérifiée, en comparant la signature existante dans le message avec une autre calculée par le nœud récepteur. Les éventuelles attaques de rejeux *Replay Attack* seront évitées par l'introduction d'un indicateur temporel *Timestamp* lors de l'envoi du message.

Pour chaque message HELLO ou TC généré, un message de signature est aussi généré dans le format de la figure ci-dessous selon les étapes suivantes :

- le nœud crée le message de routage OLSR (HELLO, TC).

- l'indicateur temporel est généré (*Actual Timestamp*).

- le message de signature est généré en utilisant le message du protocole de routage et de l'indicateur temporel actuel. À noter que les valeurs de TTL (*Time to live*) et le nombre de sauts (*Hop count*) sont mis à 0 pour éviter de générer le même message à chaque fois que ces valeurs changent (passage d'un nœud à un autre).

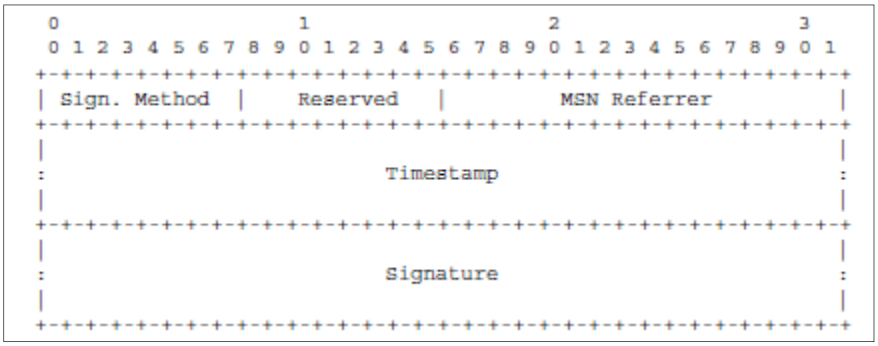

Figure 2.3: Format du message de signature généré
Tirée de Adjih et al (2003)

L'implémentation d'une telle solution exige un niveau d'énergie considérable pour calculer et vérifier les messages TC et HELLO sachant que ces deux messages sont envoyés périodiquement en permanence. De plus la solution exige la création d'un nouveau message associé à chaque message HELLO ou TC ce qui affecte et dégrade les performances (traitement supplémentaire, codage/décodage du message) du réseau et diminue sa durée de vie.

L'architecture précédente a été renforcée par une infrastructure de distribution de clés de chiffrement publiques (*Public Key Infrastructure*) (Adjih et al. 2005b) pour s'assurer de

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

l'authentification des messages signés. Ainsi les attaques d'usurpation d'identité (nœuds illégitimes se faisant passer pour des nœuds légitimes) peuvent être évitées. Toutefois les exigences en puissance de calcul cryptographique reste toujours un obstacle devant une implantation pratique de cette solution.

Raffo et al. ont proposé dans le cadre du même projet (Raffo et al. 2004) une amélioration de l'architecture de sécurité proposée par Adjih et al. (2003). L'amélioration consiste à proposer un mécanisme additionnel pour contrer les éventuels attaques d'usurpation de lien que le mécanisme d'Adjih et al. (2003) ne couvre pas dans sa proposition. En effet, le mécanisme PKI de Adjih et al. (2003) n'est pas capable de détecter un nœud compromis. Un nœud compromis même s'il est authentifié peut envoyer des messages erronés pour altérer la table de routage. Ce nœud compromis ne sera pas détecté car il possède les clés d'authentification requises pour envoyer un message valide.

L'idée du mécanisme proposé par Raffo (Raffo et al., 2004) consiste à envoyer une preuve du lien déclaré dans le message TC ou HELLO entre deux nœuds. En d'autre termes, si le nœud A envoie un message TC qui inclue le nœud B dans la liste des *MPR Selector*, le nœud A doit joindre aussi la preuve (le message HELLO) avec laquelle le nœud B a réellement sélectionné A comme MPR. Les messages sont tous signés et validés. De cette manière, même si le nœud est compromis, il ne pourra pas envoyer des messages HELLO ou TC falsifiés. Toutefois la complexité de cette solution rend son implantation un vrai défi surtout en considérant les exigences en matière de ressources requises pour le fonctionnement de ce protocole.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Système basé sur la cryptographie asymétrique

Finalement, Fourati et Agha ont proposé (Fourati et Agha, 2006) une méthode pour assurer l'intégrité des messages TC avec un mécanisme basé sur la cryptographie asymétrique. L'optimisation proposée consiste à exploiter la diffusion des messages TC à travers le réseau pour envoyer les clés publiques d'un nœud faisant parti du réseau. En procédant ainsi on peut

assurer que l'intégrité des messages envoyés est préservée, en empêchant les nœuds intermédiaires d'altérer les messages TC.

Le mécanisme de Fourati et Agha n'assure pas l'authentification ni la confidentialité, comme il ne traite pas l'intégrité des messages HELLO ni le cas d'un nœud compromis (attaque interne).

2.2.3.2 Nouveaux protocoles

Les solutions de sécurité basées sur la cryptographie peuvent certes contrecarrer les attaques externes lancées par des nœuds qui ne sont pas authentifiés. Toutefois de telles solutions ne peuvent pas détecter les attaques lancées par des nœuds internes compromis. Ces nœuds peuvent rediriger le trafic, altérer le contenu des messages ou même supprimer leur contenus d'où la difficulté de sécuriser le protocole OLSR dans sa version actuelle. Ce constat a poussé les chercheurs (Abdellaoui et Robert, 2009) à modifier certaines spécifications du protocole OLSR pour satisfaire les exigences de sécurité.

SU-OLSR (R. Abdellaoui et Robert, 2009)

Les relais MPR sont l'optimisation majeure que le protocole OLSR a apportée. Le rôle important qu'ils occupent dans les processus de routage (relais des paquets TC et données) du protocole OLSR peut inciter les nœuds malveillants à jouer ce rôle. En effet, un nœud malicieux peut chercher à présenter des bons critères pour qu'il soit sélectionné comme MPR. Une fois sélectionné, il peut exercer des attaques plus élaborées ayant plus d'impact sur le réseau vu la quantité du trafic passant par lui.

L'objectif de SU-OLSR consiste à modifier les spécifications du protocole OLSR pour y incorporer des mécanismes de sécurité afin d'assurer son bon fonctionnement malgré la présence de nœuds malicieux. Les mécanismes proposés consistent à modifier le processus avec lequel les nœuds MPR sont sélectionnés. De telle manière à éviter de sélectionner les

nœuds malveillants qui ont fait de fausses déclarations de ressources pour forcer leur sélection.

2.2.3.3 IDS (*Intrusion Detection System*)

Les solutions IDS sont proposées pour détecter les attaques internes. Ce sont les attaques que les solutions cryptographiques ne peuvent pas détecter. En effet, les attaques internes sont des attaques lancées par des nœuds compromis. Un IDS est souvent utilisé comme une seconde ligne de défense après les systèmes de cryptographie.

IDS basé sur les procédures d'investigation

Zhang (Zhang et al. ,2010) a proposé un système de détection d'intrusion organisé en grappes pour le protocole OLSR afin de contrer les attaques d'usurpation et suppression de liens. Le système propose que le système IDS soit implanté au niveau de chaque chef de grappe. De cette façon, toute anomalie détectée par les nœuds de la grappe est rapportée vers lui pour une prise de décision et une réponse adéquate.

Le système proposé imite le fonctionnement d'une cour pénale. Chaque nœud de la grappe qui détecte une anomalie, envoie une requête adressée au chef de grappe pour qu'il puisse faire des *investigations* au près du nœud suspecté. Une fois la procédure d'investigation lancée, le nœud suspecté doit se défendre en envoyant une copie du message HELLO ou TC qui prouve son *innocence*. Si la preuve d'innocence est validée par le chef de grappe, le nœud qui a lancé le processus d'investigation est sanctionné. À noter que les messages d'investigation, d'accusation et défense sont des messages signés pour contrer d'éventuelle usurpation d'identité. Le système proposé est certes original. Toutefois, il est complexe et le nombre de messages échangés affectera surement les performances du système surtout avec une éventuelle attaque de dénie de service causée par de fausses accusations.

IDS basé sur les spécifications du protocole OLSR

Wang a proposé une autre approche pour la conception d'un IDS pour le protocole OLSR (Wang et al., 2005). En effet, l'approche adoptée vise à concevoir un IDS à partir des spécifications du protocole OLSR. Dans cette configuration tous les nœuds participent à la surveillance du réseau. Chaque nœud possède un ensemble de règles de bonne conduite qu'un paquet doit satisfaire pour le valider. Autrement, le nœud moniteur diffuse une alarme dans le réseau qualifiant le nœud émetteur (ou un nœud relai) du paquet comme malicieux. Les règles de bonne conduite sont extraites à partir des spécifications du protocole OLSR pour détecter toute contradiction avec ces spécifications. Nous utilisons cette piste dans ce projet de recherche.

Adnane a proposé une extension de l'approche de Wang (Adnane et al., 2008). La nouvelle proposition consiste à construire et valider la table de routage à partir d'un modèle basé sur la confiance implicite.

L'idée principale qu'Adnane et ses collaborateurs ont apportée se base sur deux facteurs : le test et la confiance. Un nœud ne fera pas confiance à son voisin (l'ajout dans sa table de routage, liste des voisins,…) qu'après avoir vérifié que son voisin se comporte bien selon les spécifications du protocole OLSR. Autrement, le voisin sera rejeté. Le test consiste à envoyer un message HELLO à son voisin et voir si celui va l'inclure dans son prochain message HELLO (spécification du message HELLO RFC 3626). À l'issue de ce test, une relation de confiance s'installe entre les deux nœuds. À noter que les nœuds MPR sont sélectionnés à partir de la liste des nœuds de confiance pour éviter les nœuds compromis.

2.3 Vulnérabilités et type d'attaque du protocole SC-OLSR

SC-OLSR par sa nature hérite des vulnérabilités du protocole OLSR. Cependant, il introduit de nouvelles vulnérabilités puisqu'il offre de nouvelles fonctionnalités. Ce qui expose le protocole à des attaques menaçant son fonctionnement puisqu'il ne spécifie aucune

recommandation pour sécuriser son fonctionnement. Les nouvelles vulnérabilités introduites sont relatives à chaque rôle qu'un nœud occupe dans le réseau (Électeur, MPR et CH).

Nous verrons au chapitre suivant les vulnérabilités et les attaques spécifiques au protocole SC-OLSR ainsi que le système IDS proposé pour contrecarrer ces attaques.

CHAPITRE 3

ARCHITECTURE ET SYSTÈME DE SÉCURITÉ PROPOPOSÉ POUR SC-OLSR

Dans ce chapitre, nous allons présenter le système de sécurité proposé pour détecter les comportements malicieux suivant les scénarios d'attaques présentés dans le chapitre précédent.

3.1 Motivation et objectifs

Le protocole SC-OLSR est considéré comme une version améliorée du protocole OLSR classique. L'amélioration consiste à optimiser les ressources disponibles et inciter les nœuds à participer aux différents processus collaboratifs. Cependant, comme nous avons vu dans la section 2.3, ce protocole souffre de plusieurs vulnérabilités qui pourraient être exploitées à des fins frauduleuses. Dans cette optique, nous avons conçu un système de détection d'intrusion (IDS) pour contrer les attaques ayant un impact important sur le réseau.

Notre système sera basé sur les spécifications du protocole de SC-OLSR. Une logique d'analyse sera implémentée dans chaque nœud de surveillance appelé moniteur. Ces nœuds seront capables d'analyser les paquets échangés dans leur voisinage pour détecter toute contradiction avec les spécifications qui traduit un éventuel comportement malicieux. Le système proposé suit la piste de solution que Adnane et al. (2008) ont proposée (voir section 2.2.3.3).

La logique d'analyse des nœuds est décrite sous forme d'algorithme pour détecter les attaques contre ce protocole. Chaque algorithme regroupe certaines conditions à valider pour vérifier si le nœud surveillé est malicieux ou non.

Les systèmes de détection d'intrusion présentés dans la littérature sont basés sur des solutions de surveillance (monitoring) qui ne prennent pas en considération la consommation énergétique des nœuds. Par conséquent la durée de vie du réseau est affectée par la consommation des ressources exigées par ces systèmes.

3.2 Revue de littérature des solutions utilisées dans les systèmes de détection d'intrusion dans les réseaux Ad-Hoc.

3.2.1 Solution basée sur la surveillance

Mécanisme *Watchdog* and *Pathrater*

Marti et al. (2003) ont présenté une solution pour détecter les nœuds malicieux qui suppriment les paquets (de façon sélective ou aléatoire) passant par ce nœud de transit. Cette solution nommée *Watchdog* consiste en effet, à surveiller le comportement de tous les nœuds d'une part, et choisir la route la plus sécuritaire grâce au module nommé *Pathrater* d'une autre part. De ce fait, tous les nœuds du réseau se surveillent les uns les autres sous forme d'architecture maillée.

La figure 3.1 ci-dessous présente le mécanisme *Watchdog*. En effet, si le nœud S veut transmettre un paquet vers le nœud D via les nœuds intermédiaires A, B et C, le paquet est transmis au nœud A qui le retransmet à son tour au nœud B mais garde une copie du paquet. La prochaine étape du processus est de surveiller si B va retransmettre ce paquet vers le nœud C en écoutant et en comparant tous les paquets émis par le nœud B. Si le nœud B ne retransmet pas le paquet au bout d'un certains temps, un compteur est incrémenté. Si le compteur atteint une valeur maximale préétablie (nombre de fois que le nœud B ne transmet pas un paquet), le nœud A peut conclure que le nœud B est malicieux. Sa décision est rapportée au nœud S.

Les nœuds surtout de faible énergie sont condamnés à dépenser rapidement leurs ressources disponibles.

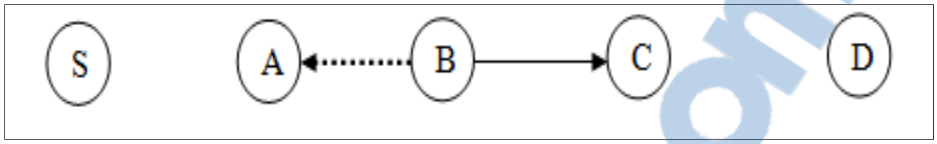

Figure 3.1 : Principe *Watchdog* de surveillance Tirée de Marti (2003)

La solution *Watchdog* ne favorise donc pas l'optimisation de ressources. Cela pourrait donc engendrer des comportements égoïstes de la part des nœuds ayant un niveau d'énergie trop bas. Ces derniers ne vont pas dépenser leur énergie pour protéger les plus forts.

CONFIDANT : un système basé sur la réputation

Buchegger et Le Boudec (2002) ont proposé une extension au protocole de routage DSR appelé CONFIDANT, utilisant un mécanisme similaire au mécanisme *Watchdog* et *Pathrater*. Chaque nœud observe le comportement de ses voisins. Une fois qu'un comportement malicieux est détecté, le nœud malicieux est exclu de tous les services offerts par le réseau (retransmission des paquets par exemple) et l'isole grâce à un système de réputation. Il peut alerter les autres nœuds par la diffusion d'un message d'alarme.

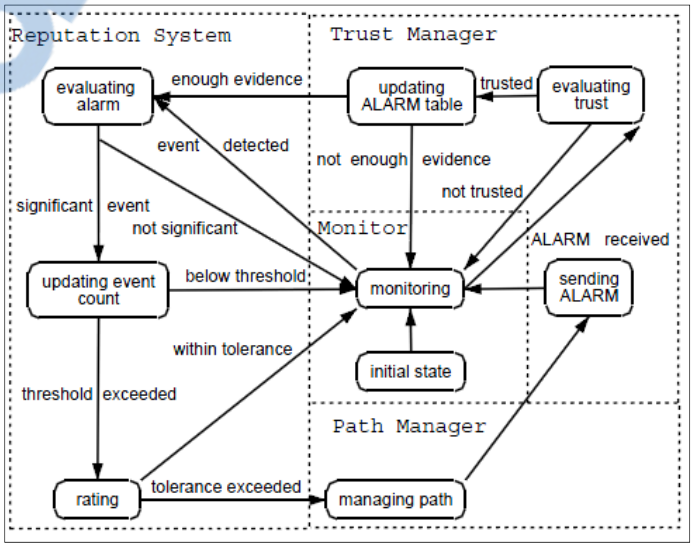

Figure 3.2: Mécanisme de détection CONFIDANT
Tirée de Buchegger et Le Boudec (2002)

Rapport-Gratuit.com

Le mécanisme proposé utilise le module *Monitoring* pour détecter toute activité malicieuse. Si un cas suspect est détecté, le module *Monitor* envoie une notification au module *Reputation System* qui, à son tour, fait une mise à jour de sa table de réputation en fonction des rapports d'activités reçus. Si la valeur de réputation dépasse un seuil critique, une alarme est envoyée aux autres nœuds via le module *Trust Manager* ainsi qu'au *Path Manager* qui supprime toutes les routes contenant le nœud malicieux.

Puisque ce protocole permet l'envoi d'alarmes, le réseau peut être sujet à des attaques envoyant de fausses accusations. Ainsi, un dénie de service peut être facilement réalisé.

CORE : un système basé sur la réputation

Le mécanisme CORE (Michiardi et Molva, 2002) a proposé une solution pour contrer le comportement égoïste des nœuds. La solution consiste à offrir des moyens d'incitation pour tout nœud voulant participer aux processus collaboratifs. Les moyens d'incitation sont inspirés de la théorie des jeux. Chaque nœud a une réputation à mettre en jeu traduisant son honnêteté. Pour transmettre ou recevoir un paquet, le nœud doit avoir une réputation suffisante. De plus, chaque nœud détecté malveillant ou égoïste voit sa réputation diminuer ce qui a pour conséquence d'isoler ce nœud complètement du réseau (impossibilité d'émettre ou recevoir des paquets). Cela oblige les nœuds rationnels à adopter un comportement honnête.

Dans CORE, chaque nœud attribue une valeur de réputation à tout autre nœud impliqué dans un processus coopératif. À noter que CORE contrairement à CONFIDANT ne permet d'attribuer que des valeurs positives pour la réputation, si le nœud de décision reçoit un rapport positif d'un autre nœud (surveillance indirecte). Les valeurs négatives sont réservées seulement pour une surveillance directe dans le cas où le nœud local surveillé ne coopère pas. En procédant ainsi, le mécanisme élimine les éventuelles fausses accusations et les attaques de dénis de service dont CONFIDANT souffre. Si le nœud A sollicite un service du nœud B (retransmission de paquet, découverte de routes), le nœud B consulte sa table de réputation et calcule la valeur globale de réputation (surveillance directe et indirecte) pour le nœud A. S'il

s'avère que le nœud A a une réputation globale négative alors sa requête sera rejetée et le nœud sera isolé.

3.2.2 Solution IDS pour les réseaux Ad-Hoc

L'expérience a montré que l'utilisation des techniques de prévention d'intrusion comme la cryptographie ou l'authentification n'est pas suffisante pour contrer les attaques internes. Toutefois, ce type de système est utilisé comme première ligne de défense alors que la deuxième ligne de défense est occupée par les systèmes de détection d'intrusion communément désignés par son acronyme anglais IDS (*Intrusion Detection System*). Le principe général d'un IDS est composé de trois phase : une phase de collection de données suivie d'une phase d'analyse et enfin une phase de réponse pour prévenir ou minimiser l'impact sur le système. Le système IDS est implanté au niveau de certains nœuds spéciaux appelés moniteurs ou nœuds de surveillance. Le déploiement de ces nœuds diffère en fonction du type protocole et de l'architecture de l'IDS.

Les IDS peuvent être classifiés selon la façon avec laquelle les données sont collectées à savoir : IDS basé sur les données collectées du réseau (*Network-Based IDS*) ou IDS basé sur l'analyse des fichiers de trace et données du système d'exploitation (*Host-Based IDS*). On peut aussi classer les IDS selon les techniques de détection utilisée.

- **système de détection d'anomalie** : le système détecte tout comportement qui dévie du comportement normal préétabli et déclenche une réponse.

- **système basé sur les signatures** : le système possède une base de données du comportement de certaines attaques (fichier dll ciblés, port utilisé,…) avec laquelle sont comparées les données collectées. Une attaque est détectée si les données collectées coïncident avec un comportement malicieux déjà enregistré.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

- **système basé sur les spécifications** : le système défini un ensemble de conditions qu'un programme ou un protocole doit satisfaire. Une attaque est détectée si le programme ou le protocole ne respecte pas les conditions établies du bon fonctionnement.

L'architecture idéale d'un IDS dépend de l'infrastructure elle-même. Il existe différentes architectures proposées dans la littérature à savoir : autonome, hiérarchique ou basée sur des agents mobiles selon Mishra et al. (2004).

Comme on va voir dans le reste de cette section, plusieurs architectures d'IDS ont été proposées pour présenter un système de détection d'intrusion capable de satisfaire les exigences complexes des réseaux Ad-Hoc.

IDS coopératif et distribué

Zhang et Lee (2000) ont proposé une architecture distribuée coopérative où chaque nœud participe dans le processus de détection. Chaque nœud, appelé agent IDS, est responsable de la collection des données et la détection des activités malicieuses. Chaque agent IDS peut initier une réponse (punition) indépendamment des autres nœuds. Toutefois, les agents IDS voisins pourraient coopérer entre eux pour une détection d'intrusion globale.

Le modèle de l'agent IDS est structuré en six modules :

- le module *local data collection* collecte les données en temps réel incluant les événements du système et les opérations faites par l'utilisateur.

- le module *local detection engine* decide à partir des données collectées si le système est attaqué ou non. Le module peut initier une réponse si l'attaque est détectée avec des preuves précises. La réponse est exécutée par le module *local response* (alerte à l'utilisateur local) ou le module *global response* (alerte globale) en fonction du type d'attaque de protocole ou d'application utilisée.

- le module *cooperative detection engine* est exécuté quand une anomalie est détectée avec des preuves faibles et sollicite la coopération des autres nœuds via un autre module de communication sécurisée appelé *secure communication.*

IDS coopératif pour les protocoles organisés en grappe

Huang et Lee (2003) ont adopté le même mécanisme proposé par Zhang et Lee (2000) mais appliqué à des réseaux organisés en grappes. Le chef de grappe est élu de façon aléatoire. Il se charge de protéger sa grappe en analysant tout paquet passant par lui vers un membre de cette grappe.

L'IDS proposé peut non seulement détecter une intrusion mais aussi déterminer la source de l'attaque via des outils statistiques. Le système est évalué périodiquement via ces outils pour analyser le trafic du réseau et les activités du système. Une attaque est détectée si les statistiques varient de celles pré-calculées (système de détection d'anomalie).

Les statistiques utilisés sont classés en deux catégories : la première catégorie sont les données collectées sur la mobilité des nœuds et les fichiers de traces enregistrés indépendamment par chaque nœud, et la deuxième catégorie sont les statistiques basées sur l'analyse de trafic utilisant les données collectées sur le routage, la retransmission des paquets et le nombre de paquets reçus/transmis.

Ce modèle propose une nouvelle approche pour créer un point central de protection comme dans les réseaux filaires. Mais il ne traite pas le cas d'un éventuel chef de grappe malicieux ou compromis qui pourrait avoir un lourd impact sur la sécurité du réseau.

Mohammed et al. (2008) ont proposé une approche pour améliorer le travail précédent. L'amélioration consiste à proposer une nouvelle méthode d'élection de chef de grappe. Cette méthode prend en charge l'énergie résiduelle. Le mécanisme proposé est basé sur le système d'incitation pour encourager les nœuds à participer dans le processus de détection.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

L'objectif de l'approche proposée est de balancer la consommation d'énergie à travers tout le réseau. Les nœuds ayant suffisamment d'énergie (chef de grappe) vont servir comme IDS pour protéger les autres nœuds. Toutefois, l'existence de certains nœuds égoïstes empêche le déroulement de ce processus coopératif. L'idée donc est de forcer les nœuds à déclarer leurs vraies ressources et éviter de les cacher en offrant des incitations sous forme de réputation. En procédant ainsi les auteurs estiment que la durée de vie du réseau est prolongée. Notre solution suit cette piste en modifiant légèrement le modèle proposé comme mentionné dans (Chriqi, Otrok et Robert, 2009).

3.3 Modèle d'attaque contre le protocole SC-OLSR

Le modèle d'attaque du protocole SC-OLSR est basé sur l'analyse des spécifications du protocole. Le modèle d'attaque présenté traite uniquement les attaques qui menacent la manipulation frauduleuse des messages de routage (messages HELLO et TC) et non la transmission des paquets en générale.

Chaque nœud du réseau selon le protocole SC-OLSR joue un rôle précis dans sa grappe (Électeur, CH ou MPR). Chaque nœud a des responsabilités à assumer pour assurer le bon fonctionnement du réseau. En fonction du rôle occupé, le comportement malicieux d'un attaquant aura plus ou moins d'impact.

Cas 1 : un nœud Électeur malicieux

Pour un électeur malicieux, la seule possibilité d'effectuer une attaque est d'empoisonner les messages ELECTION ou HELLO avec des informations erronées :

Attaque 1 : **vote pour le nœud le plus faible**

Dans cette attaque, l'électeur malicieux vote pour le voisin qui a le niveau d'énergie le plus bas pour qu'il soit son chef de grappe (CH). Sachant qu'un CH joue aussi le rôle d'un MPR

spécialisée (tâche consommant beaucoup de ressources), le niveau d'énergie de cette victime diminuera donc rapidement. Cette attaque a pour conséquence de diminuer la durée de vie des nœuds faibles causant ainsi un déni de service.

Dans la figure ci-dessous, le nœud malicieux a choisi le nœud le plus faible (donc ayant le CTF le plus élevé) N3 comme CH au lieu de N1. Il diffuse un message ELECTION contenant l'adresse du voisin choisi comme CH. Ce message est diffusé à tous les voisins existant dans la portée radio de ce nœud comme le stipule les spécifications du protocole SC-OLSR.

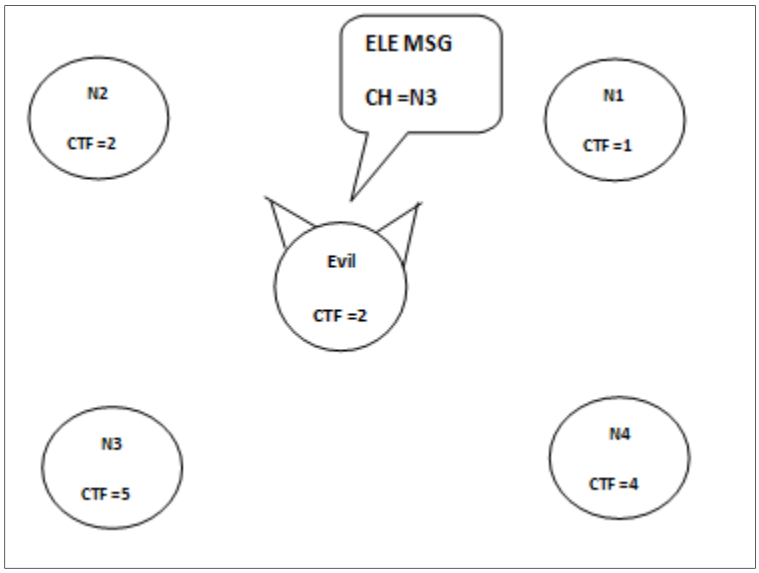

Figure 3.3: Attaque de vote pour le nœud faible comme CH

Attaque 2 : **vote pour plusieurs chefs de grappe**

Cette attaque est considérée comme une variante de l'attaque précédente avec un impact plus important. En effet, l'électeur malicieux vote pour plusieurs nœuds faibles en même temps ce qui cause des congestions de liens et un déni de service rapide. Réduisant ainsi la durée de vie du réseau.

Dans la figure 3.4, le nœud malicieux choisit plusieurs nœuds comme CH dans un intervalle de temps court. Il diffuse des messages ELECTION pour l'annoncer aux voisins comme le stipule les spécifications de SC-OLSR.

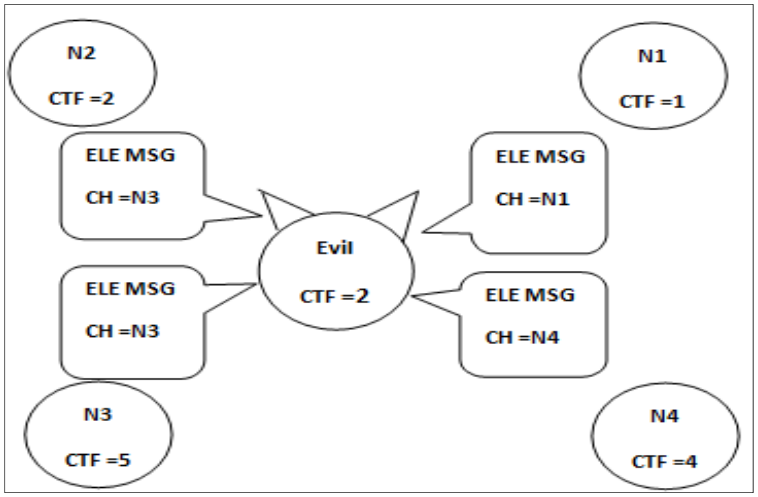

Figure 3.4: Attaque de vote pour plusieurs CH

Attaque 3 : **suppression des nœuds voisins**

Un nœud électeur malicieux pourrait ne pas déclarer un voisin dans son message HELLO pour l'isoler du réseau et causer un déni de service.

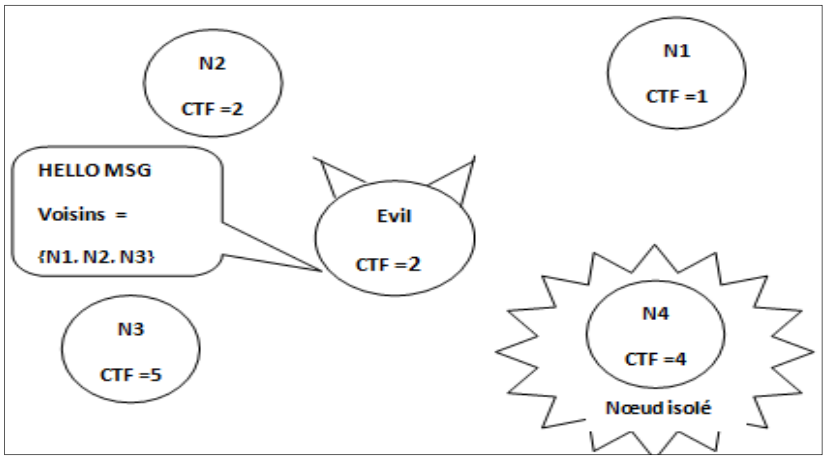

Figure 3.5: Attaque de suppression de voisins dans message HELLO

Dans la figure 3.5, malgré que le nœud N4 a voté pour le nœud malicieux, celui-ci ne l'inclue pas dans la liste des voisins de son message HELLO. Donc ce nœud sera isolé du reste du réseau si le nœud malicieux est son seul voisin.

Attaque 4 : **usurpation d'identité d'un nœud valide**.

Un nœud malicieux peut voter pour un nœud CH légitime utilisant une fausse identité. Par conséquent, le CH diffuse dans le réseau cette fausse information par un message TC et empoisonne la table de routage avec de fausses informations.

Cas 2 : un chef de grappe (CH) malicieux

Le nœud CH est considéré comme la pièce maitresse d'une grappe. Il sélectionne les MPR et doit retransmettre les paquets vers un nœud qui a voté pour lui. S'il advient qu'un CH soit compromis, les attaques lancées par ce nœud auront un impact important sur les performances du réseau.

Attaque 1 : **ajout d'un nœud électeur non légitime**

Un CH a le devoir de connecter les nœuds qui ont voté pour lui (les nœuds de la grappe) au reste du réseau moyennant les messages TC. Un nœud CH malicieux pourrait exploiter cette propriété à des fins malicieuses. En effet, le CH malicieux pourrait usurper un lien en ajoutant dans son message TC (qui contient la liste des électeurs) un nœud victime dans le réseau qui n'a pas voté pour lui. Cette attaque a pour conséquence non seulement de modifier la topologie du réseau, mais aussi de rediriger tout le trafic de la victime vers le chef de grappe malicieux comme le montre la figure 3.6. Dans cette figure, le nœud N4 a voté pour le voisin NX comme son chef de grappe (nœud présentant les meilleurs critères). Toutefois le nœud malicieux a ajouté le nœud N4 dans le message TC comme sélectionneur, créant ainsi un lien électeur-chef de grappe illégitime. Ce lien sera diffusé à travers le réseau (via des MPR) et causera des empoisonnements dans la table de routage de tous les nœuds qui vont

recevoir le message TC émis par le nœud malicieux. Ceci a pour conséquence de rediriger le trafic destiné au nœud N4 vers le nœud malicieux.

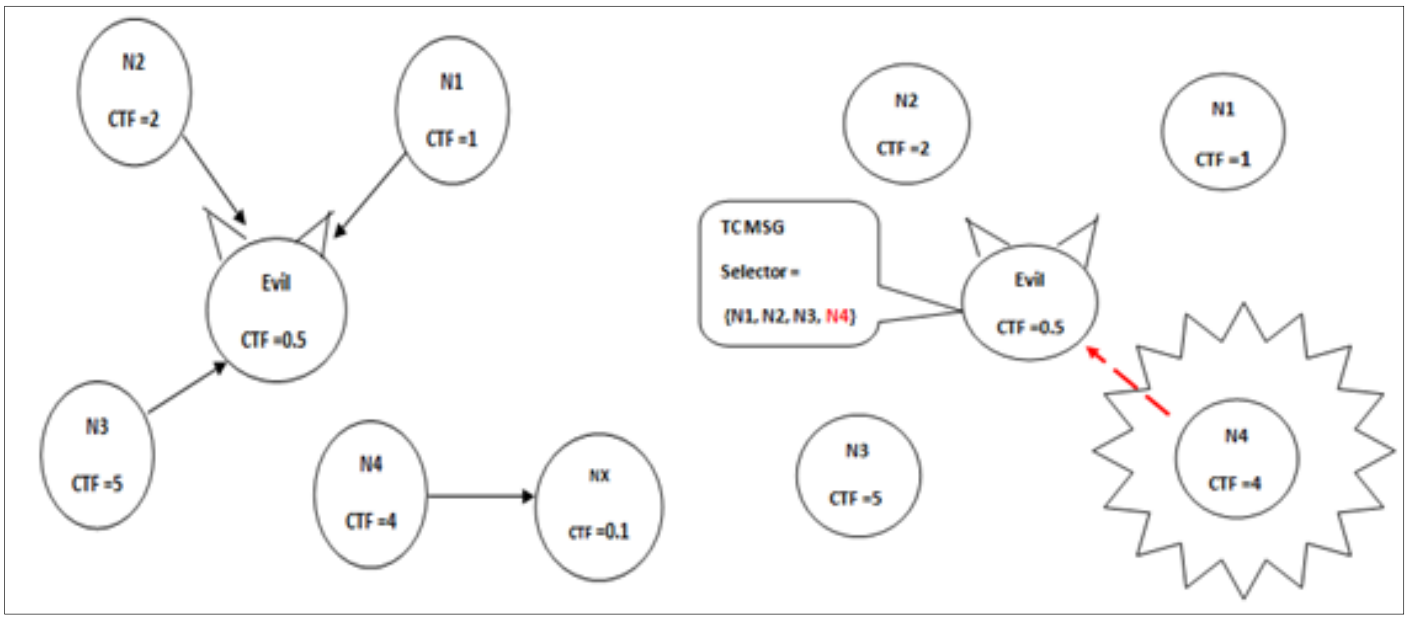

Figure 3.6: Attaque d'ajout de sélectionneur illégitime

Attaque 2 : **suppression d'un nœud sélectionneur légitime**

À l'opposé de l'attaque précédente, le CH malicieux va supprimer un nœud électeur légitime de la liste des *MPR Selector* du message TC. En procédant ainsi le nœud victime sera isolé du reste du réseau.

Dans la figure 3.7, le nœud malicieux n'a pas inclus le nœud N4 dans le message TC malgré que ce dernier a voté pour lui comme chef de grappe. Ceci isole la victime du reste du réseau.

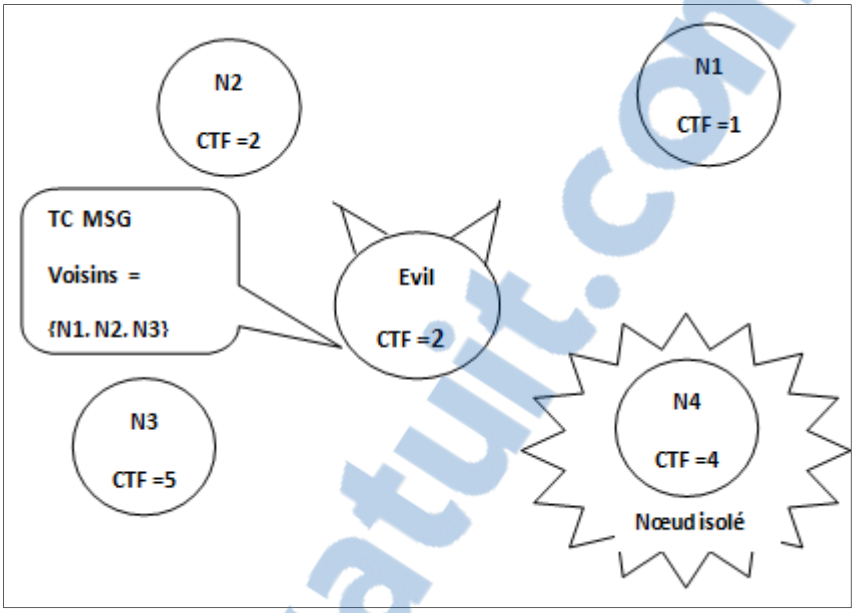

Figure 3.7 : Suppression d'un nœud sélectionneur légitime

Cas 3 : un nœud relais (MPR) malicieux

Un nœud MPR malicieux exerce les mêmes attaques qu'un CH malicieux puisqu'ils ont tous les deux les mêmes rôles (transmission/retransmission des messages TC). À noter que la liste des nœuds dans un message TC transmise par un nœud MPR contienne uniquement des nœuds élus comme chefs de grappe. Par conséquent, les nœuds ciblés par des nœuds MPR malicieux sont les CH qui seront sujets à des attaques Ajout/Suppression de nœud comme décrit précédemment. Toutefois, nous estimons que l'impact d'une attaque ciblant les nœuds élus comme CH est plus important.

3.4 Algorithme de détection

3.4.1 Algorithme de détection d'un électeur

Les algorithmes de détection proposés sont basés sur les spécifications du protocole SC-OLSR. Les nœuds choisis comme IDS ont la capacité d'analyser les paquets échangés afin de détecter tout éventuel comportement malicieux.

Rapport-Gratuit.com

Les algorithmes de détection présentés suivront la structure suivante :

Tableau 3.1: Structure des algorithmes de détection d'attaques

Hypothèse **Si** le nœud est choisi comme IDS (moniteur) **alors** **Si** paquet analysé est en contradiction avec les spécifications **alors** **Conclusion**

Algorithme 1 : Détection de l'attaque vote pour le nœud le plus faible

Tableau 3.2: Critère de détection de l'attaque vote pour le nœud le plus faible

Soit $n \in N_1(m)$ un nœud moniteur. **si** il existe un nœud $B \in N_1(m) \cap N_1(n)$ tel que : • Le nœud *m* a voté pour *A* • *B* présente de meilleurs critères d'élection **alors** *n* peut détecter le nœud malicieux *m*.

Pour élire un chef de grappe, le nœud doit émettre un message ELECTION à tous ses voisins. Un nœud peut détecter cette attaque en combinant les messages HELLO (pour vérifier s'il n'existe pas un nœud présentant de meilleurs critères) et le message ELECTION (pour identifier le CH victime) s'il est voisin seulement du nœud malicieux.

Dans la figure 3.8, le nœud moniteur *n* peut détecter le nœud malicieux *m* à condition qu'il soit voisin aux nœuds *m* et *B* pour «écouter» et corréler les messages HELLO et les messages ELECTION de ces deux nœuds.

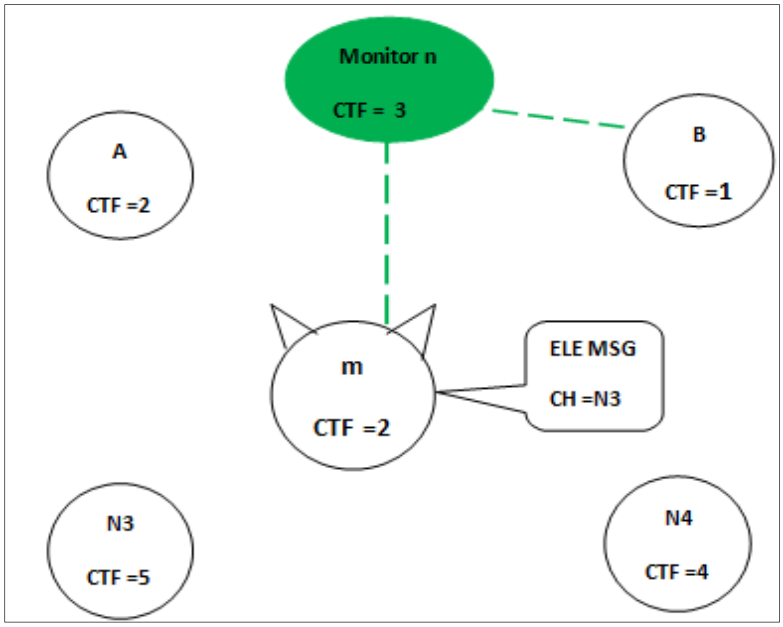

Figure 3.8 : Détection de l'attaque vote pour le nœud faible

Algorithme 2 : Détection de l'attaque vote pour plusieurs CH

Tableau 3.3: Critère de détection de l'attaque vote pour plusieurs CH

Soit $n \in N_1(m)$ un nœud moniteur Supposons que *m* vote pour $A_1, \ldots, A_n$ dans un bref intervalle **si** l'une des conditions est remplie : • Critère 1 : Vote pour le nœud le plus faible • Critère 2 : L'intervalle de temps d'élection est trop bref **alors** *n* peut détecter le nœud malicieux *m*.

Chaque nœud devrait choisir un nœud unique pour servir comme CH. S'il s'avère qu'un nœud malicieux a voté pour plusieurs CH pendant un intervalle de temps inférieur à l'intervalle d'émission des messages ELECTION alors ce nœud est considéré comme malicieux.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

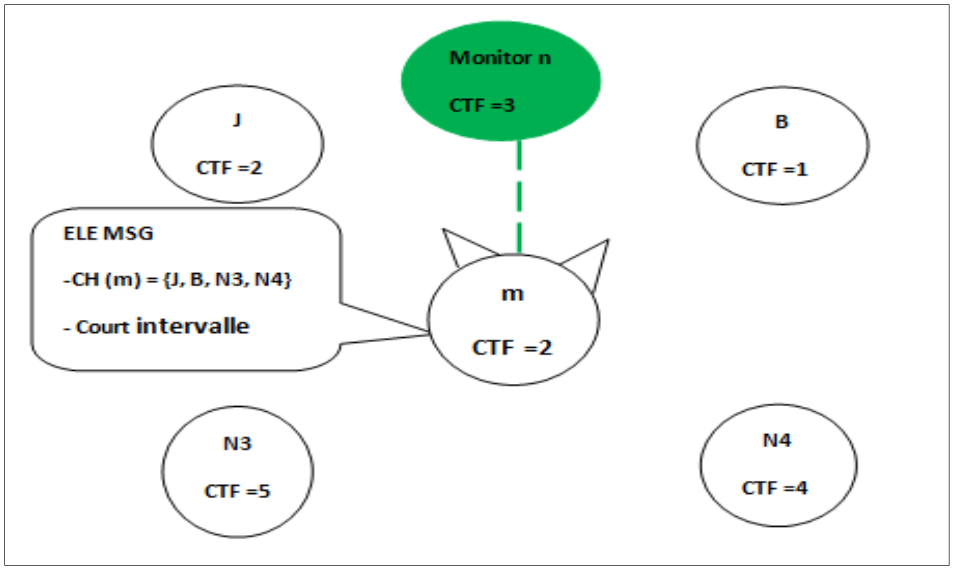

Figure 3.9 : Détection de l'attaque de vote pour plusieurs CH

Algorithme 3 : Suppression d'un nœud légitime

Chaque nœud devrait diffuser un message HELLO contenant tous les voisins à un saut. Tout nœud malicieux supprimant un nœud de sa liste dans le message HELLO peut être détecté par le critère illustré dans le tableau 3.4.

Ce critère stipule que cette attaque pourrait être détectée par un nœud moniteur qui doit être un voisin de la victime et du nœud malicieux à la fois. À noter que la victime peut aussi détecter cette attaque (puisqu'il reçoit aussi le message HELLO modifié).

Tableau 3.4 : Critère de détection de l'attaque suppression d'un voisin existant dans le message HELLO

Soit $n \in N_1(m)$ un nœud moniteur

si il existe un nœud *m* et $p \in N_1(n)$ tel que :

- $m \in N_1(p)$ et p $\notin N_1(m)$

alors *n* peut détecter le nœud malicieux *m*.

Dans ce cas, cette détection pourrait engendrer des fausses détections et accusations. Un nœud pourrait prétendre être victime de cette attaque pour sanctionner un autre nœud.

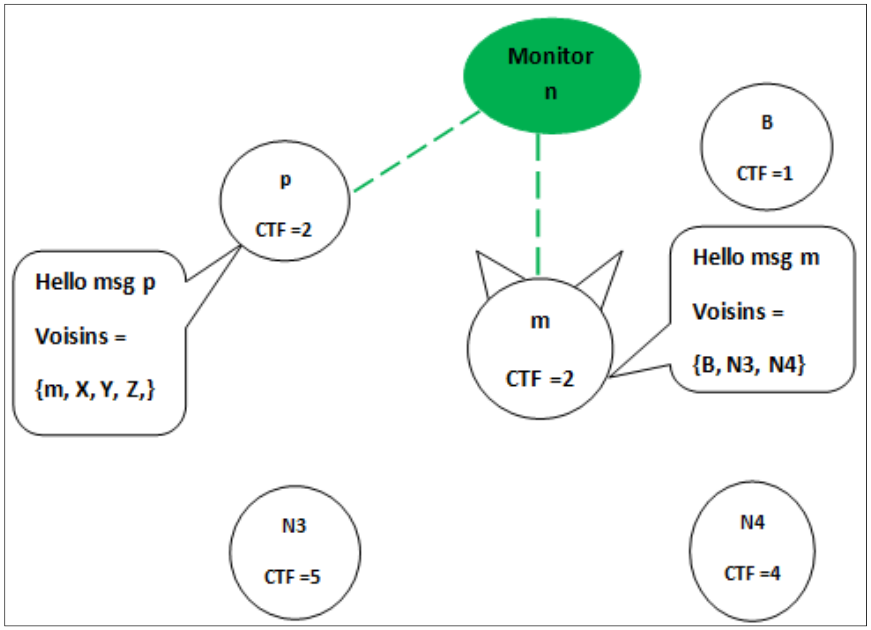

Figure 3.10: Détection de l'attaque suppression
d'un voisin dans le message HELLO

3.4.2 Algorithmes de détection d'un CH/MPR

Une fois sélectionnés, les CH/MPR sont responsables de diffuser les informations sur la topologie et transmettre le trafic vers les destinations (les nœuds qui ont sélectionné les CH ou MPR). Les nœuds CH et MPR ont un rôle similaire dans les spécifications du protocole SC-OLSR. Leurs modes d'attaque se ressemblent aussi. Par conséquent, les algorithmes de détection peuvent s'appliquer à la fois à un nœud CH ou MPR malicieux et les nœuds victimes sont appelés des sélectionneurs ou *Selectors*.

Algorithme 4 : Suppression d'un sélectionneur

Un nœud peut sélectionner un voisin comme CH ou MPR. Ce dernier doit envoyer à tous les nœuds du réseau la liste des nœuds qui l'ont sélectionné via un message TC. Par conséquent,

cette attaque peut être détectée en analysant les messages d'élection/sélection. L'algorithme se formule comme suit :

Tableau 3.5 Critère de détection de l'attaque suppression d'un nœud sélectionneur

Soit $n \in N_1(m)$ un nœud moniteur Supposons que m a sélectionné le nœud p, comme CH ou MPR avec son message ELECTION ou HELLO alors **si** $n \in N_1(p)$ • p n'inclut pas m dans ses messages TC-ou, par extension, ne diffuse aucun message TC. **alors** n peut détecter le nœud malicieux p

Un nœud peut facilement détecter un CH ou MPR malicieux voisin s'il transmet un message qui ne respecte pas les spécifications. Cependant, ce nœud doit dépenser de l'énergie pour analyser tous les messages émis pas tous les nœuds de la grappe. Étant donné que les messages TC sont retransmis par les CH ou MPR, les conditions de détection peuvent être moins contraignantes. Si tous les nœuds retransmettent des messages TC non altérés et que tous les messages envoyés par le nœud p ne contiennent pas le nœud victime, alors le nœud moniteur peut conclure que le nœud émetteur du TC message est malicieux. Autrement, un des MPR sélectionnés est malicieux ce qui exige l'implication d'autres nœuds moniteurs.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

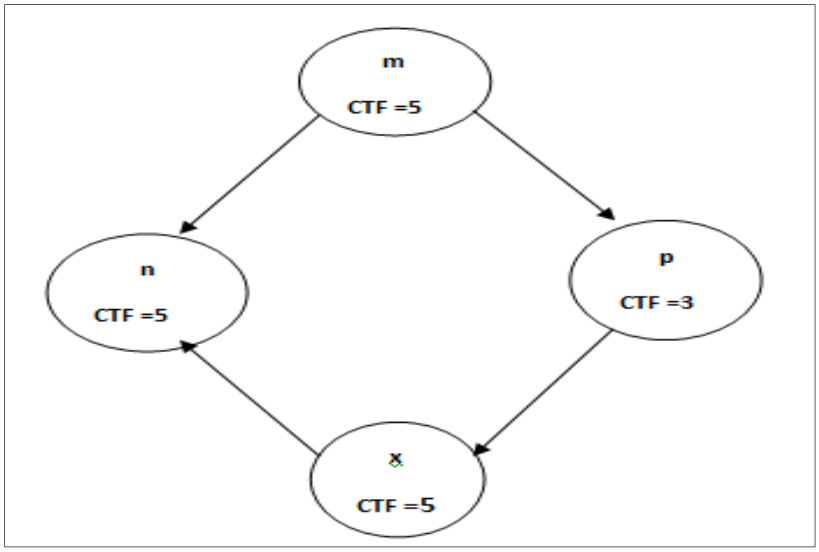

Figure 3.11: Détection à distance avec l'algorithme 4

Tableau 3.6: Critère de détection de l'attaque
suppression d'un nœud sélectionneur – Détection à distance

Soit $n \in N_1(m)$ un nœud moniteur

Supposons que m a sélectionné le nœud p, comme CH ou MPR avec son message ELECTION ou HELLO alors

si $n \notin N_1(p)$

- n reçoit des messages TC d'un voisin MPR x
- p n'inclue pas m dans ses messages TC -ou, par extension, ne diffuse aucun message TC.

alors n peut détecter le nœud malicieux p ou un nœud MPR dans $N_1(m)$.

Algorithme 6 : Ajout d'un nœud sélectionneur illégitime

Un nœud malicieux peut prétendre être sélectionné comme CH ou MPR de la victime. Cette attaque est possible si le nœud malicieux commence à émettre des messages TC avec des

fausses annonces. Un nœud voisin de la victime et du nœud malicieux peut détecter cette attaque en corrélant les messages HELLO et ELECTION avec l'algorithme suivant :

Tableau 3.7: Critère de détection de l'attaque ajout
d'un nœud sélectionneur illégitime– Détection de proximité

Soit $n \in N_1(m)$ un nœud moniteur
Supposons que *m* n'a pas sélectionné le nœud *p*, comme CH ou MPR
si $n \in N_1(p)$

- *p* inclut *m* dans les messages TC

alors *n* peut détecter le nœud malicieux *p* ou la présence d'une attaque d'usurpation d'identité sans pouvoir détecter la source.

Comme le cas précédant, les critères de détection peuvent être moins contraignants. Le nœud moniteur peut être seulement un voisin de la victime pour pouvoir détecter l'attaque.

Tableau 3.8: Critère de détection de l'attaque ajout
d'un nœud sélectionneur illégitime– Détection à distance

Soit $n \in N_1(m)$ un nœud moniteur
Supposons que *m* n'a pas sélectionné le nœud *p*, comme CH ou MPR
si $n \notin N_1(p)$

- *n* reçoit des messages TC d'un voisin MPR *x*
- *p* inclut m dans les messages TC

alors *n* peut détecter le nœud malicieux *p* ou le nœud MPR *x* dans $N_1(n)$ ou encore la présence d'une attaque d'usurpation d'identité sans pouvoir détecter la source.

Il est à remarquer que les deux derniers critères peuvent détecter des attaques d'usurpation d'identité sans pouvoir identifier la source. Toutefois en déployant plus de ressources, la

précision de détection peut être améliorée, mais démunira la durée de vie du réseau. L'attaque d'usurpation d'identité est l'une des attaques les plus complexes à détecter dans les réseaux Ad-Hoc. De futurs travaux de recherche seront consacrés à ce problème.

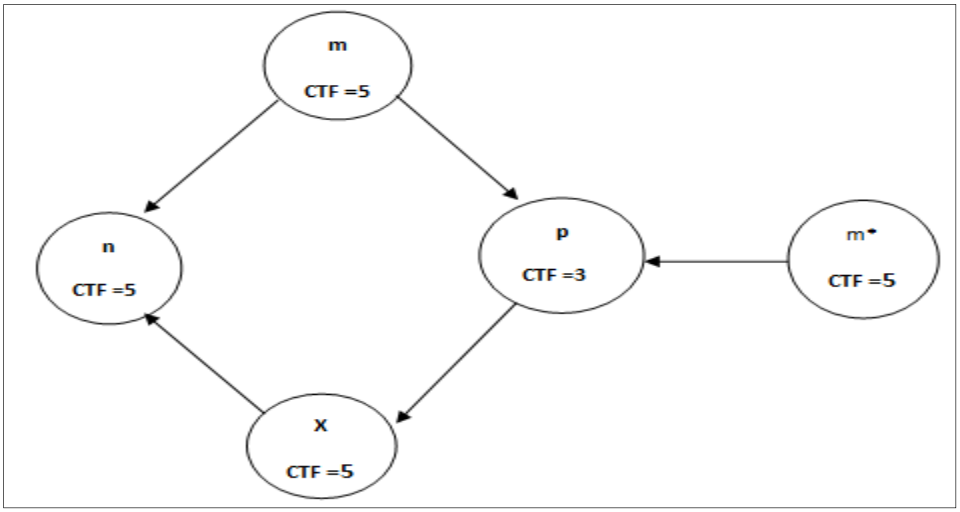

Figure 3.12: Détection à distance et attaque de clonage

3.5 Architecture et déploiement de l'IDS

3.5.1 Architecture

L'architecture proposée de l'IDS est conçue de façon modulaire afin de répondre aux besoins présents et pouvoir ajouter facilement des fonctionnalités pour les futurs besoins.

L'architecture est composée de 3 modules (voir figure 3.13) à savoir *Packet Sniffer*, *Packet Analyser* et *Packet Dispatcher*. Une fois que la décision est prise (nœud malicieux ou non), le module *Packet Analyser* doit communiquer avec un system de punition *Punishment System.* Ce système est remplacé, dans cette première version, par un système d'enregistrement de traces pour calculer la probabilité de détection ultérieurement.

- **packet sniffer**: ce module collectionne les paquets échangés dans la portée radio du nœud IDS et les envoie au ***Dispatcher.***

- **packet dispatcher**: ce module consiste à trier les paquets envoyés par le module ***Packet Sniffer*** en fonction du type de message ELECTION, TC ou HELLO.

- **packet analyser**: ce module analyse les messages extraits des paquets recueillis. Chaque message est vérifié pour s'assurer qu'il respecte les spécifications en fonction de son type et son contenu. À noter qu'il y a un sous module d'analyse pour chaque type de message ELECTION, TC ou HELLO.

Des structures de données sont mises en place pour stocker les informations au sujet des voisins (MPR, CH, voisin). Ces données sont consultées au besoin lors de l'analyse. À l'issu de l'analyse effectuée, une décision est prise concernant la nature du nœud émetteur du paquet en fonction de la conformité aux spécifications des informations extraites.

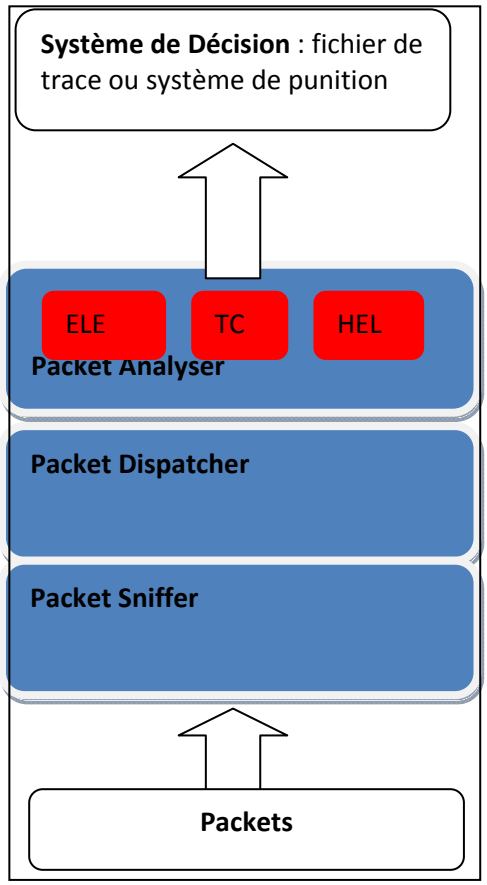

Figure 3.13 : Architecture du système de détection d'intrusion proposée

3.5.2 Déploiement

La stratégie de déploiement d'un IDS traduit la façon avec laquelle les nœuds IDS seront distribués. Cette stratégie a des conséquences directes sur l'efficacité de l'IDS plus particulièrement dans un réseau Ad-Hoc. Une stratégie efficace est une stratégie capable protéger une grande partie du réseau tout en employant des ressources minimales.

Plusieurs stratégies de déploiement d'IDS ont été proposées dans la littérature dans le contexte du protocole OLSR. Dans les récents travaux de recherche (Mohammed et al., 2008) (Zhang et al., 2010) (Huang et al., 2003) les chercheurs ont opté pour une organisation en grappe pour implémenter leur IDS. Dans cette configuration, les nœuds CH sont souvent choisis comme IDS. Cette stratégie a toutefois des inconvénients, à savoir :

- le nœud CH a une grande surcharge (relayer le trafic et protéger la grappe) qui fera décroitre son énergie rapidement.

- les activités du nœud CH dans cette stratégie ne sont pas surveillées. Il pourrait aussi être malicieux ce qui constitue une grande menace pour le réseau.

La stratégie de déploiement proposée consiste en effet à distribuer de façon aléatoire les nœuds IDS dans le réseau. En procédant ainsi, nous estimons que la majorité des nœuds seront couverts et surveillés. En plus, les adversaires ne pourront plus connaitre les nœuds IDS avec certitude ce qui complique la stratégie malicieuse à adopter par ces nœuds. À noter que les critères de construction de grappes adoptés dans les travaux antérieurs (Mohammed et al., 2008) (Zhang et al., 2010) (Huang et al., 2003) sont différents de ceux du protocole SC-OLSR ce qui complique la comparaison en terme de performance et rendement entre ces protocoles.

Rapport-Gratuit.com

Nous verrons dans le chapitre suivant les résultats de la simulation des scénarios proposés. En particulier, l'impact de la stratégie de déploiement de nœuds moniteurs sur les pourcentages de détection des nœuds malicieux.

CHAPITRE 4

Scénarios de simulation et expérimentations

4.1 Motivation et objectifs

L'objectif des expérimentations effectuées est de trouver un compromis entre la sécurité et la durée de vie du réseau. En d'autres termes, optimiser le nombre de nœuds moniteurs pour prolonger la durée de vie du réseau et en même temps offrir un niveau de sécurité acceptable. Un nombre faible de nœuds moniteurs va détecter peu d'attaques, mais il va consommer moins d'énergie.

Pour comparer l'efficacité de la solution proposée, l'approche du mécanisme *Watchdog* est utilisée comme témoin. Dans cette configuration classique, tous les nœuds participent à l'opération de surveillance du réseau. Le niveau de sécurité est élevé, mais cela diminue considérablement la durée de vie des nœuds.

Dans l'objectif de mesurer la performance de cette approche, des simulations ont été effectuées sous NS2 (*Network Simulator*) (Fall et Vardhan, 2008) et UM-OLSR (Roy, 2006) ainsi que des outils développés pour l'analyse des traces et résultats de la simulation.

4.2 Environnement de simulation

4.2.1 Network Simulator NS2

NS2 est un simulateur orienté objet, écrit en C++ avec une interface de simulation en OTCL, une extension orienté objet de TCL (*Tool command Language*). Le simulateur supporte la hiérarchie de classe en C++ et en OTCL. L'utilisateur crée une simulation à partir d'un fichier en TCL dit fichier de scénario. Ce fichier regroupe des objets de simulations (nœud, topologie, type de lien, etc.…) qui sont instanciés par un interpréteur écrit en OTCL. Le corps du simulateur gère les différents objets instanciés et orchestre le déroulement de la

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

simulation et l'interaction entre les objets. Le simulateur est écrit en C++ pour acquérir une rapidité de calcul.

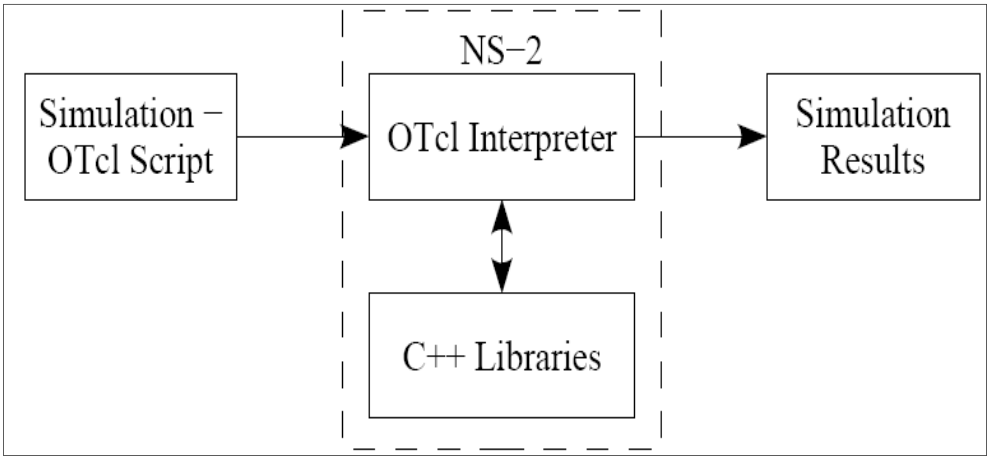

Figure 4.1: Flot de simulation de scénarios sous NS2

La visualisation graphique du déroulement du scénario est assurée par l'extension NAM (*Network Animator*) (Fall et Vardhan, 2008). L'interface NAM permet de visualiser la topologie, l'évolution du scénario et le trafic entre nœuds.

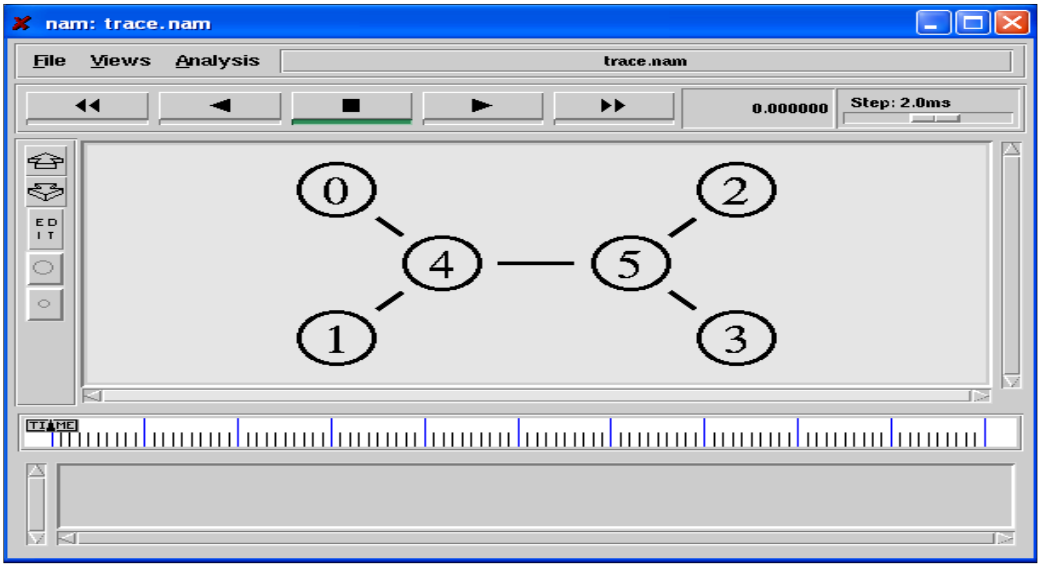

Figure 4.2 : Création de topologie sous NAM/NS2

UM-OLSR (Roy, 2006) est une extension ajoutée au simulateur NS2. Cette extension permet de simuler le protocole OLSR en respectant les exigences de la RFC-3626.

4.3 Scénarios de simulation

Les simulations ont été effectuées afin de déterminer l'impact du pourcentage de nœuds moniteurs sur la probabilité de détection en adoptant la stratégie de déploiement proposée (voir chapitre 3). Le code source du package UM-OLSR a été modifié pour supporter les exigences de SC-OLSR.

Les expérimentations ont été conduites sur des scénarios statiques pour mesurer la performance du système dans des conditions défavorables. En effet, nous pensons que les résultats seraient meilleurs dans un environnement mobile, car les nœuds moniteurs seraient en mobilité permanente ce qui devrait améliorer la probabilité de détection. Cette hypothèse devra être vérifiée dans les travaux futurs.

Dans les différents scénarios simulés, les nœuds moniteurs sont choisis de manière aléatoire. Cette approche présente des avantages par rapport au processus d'élection/sélection de nœuds moniteurs dans lequel un nœud malicieux peut forcer son voisin à le choisir comme moniteur (en annonçant des paramètres de sélection idéaux). De telles situations pourraient avoir de lourdes conséquences sur la sécurité des différents services offerts par le réseau.

La simulation commence par la distribution de façon aléatoire de tous les nœuds dans une topologie carrée. Par la suite, les moniteurs et les nœuds malicieux sont choisis de façon arbitraire pour les raisons mentionnées précédemment. Enfin, le pourcentage de nœuds malicieux est fixé 10 % ce qui va défier le système de détection et affecter la probabilité de détection. Les paramètres de la simulation sont résumés dans le tableau ci-dessous.

Des outils de paramétrage de scénarios de simulation ont été élaborés en utilisant le langage TCL/TK ou des scripts Shell.

Les scénarios sont les points d'entrée de la simulation avec lesquels les différents objets de la simulation sont crées.

Tableau 4.1: Résumé des paramètres de simulation

Modèle de propagation	TwoRayControl
Type de réseau	IEEE 802_11
Zone de simulation	Carré : 750m x 750m
Nombre total des nœuds	De 10 à 100 nœuds
Nombre de nœuds moniteurs	20 %, 50%, 80%, 100%
Nombre de nœuds malicieux	10 %
Temps de simulation	40 sec
Nombre de simulation par expérimentation	20

Les résultats des simulations sont des fichiers de traces qui enregistrent plusieurs informations et événements sur la simulation comme le nœud détecté, le type d'attaque et le nœud moniteur qui a détecté l'attaque. Ces fichiers de traces sont générés par des classes spéciales que nous avons incorporées dans le code d'UM-OLSR. Devant le nombre élevé des fichiers de traces, nous avons conçu des outils d'automatisation et d'analyse statistique pour analyser les résultats de la simulation.

Nous avons utilisé *Gnuplot* (Janert, 2008) afin de présenter graphiquement les résultats obtenus stockés dans les fichiers. Le tableau résume les outils crées et utilisé dans l'analyse.

Tableau 4.2: Outils crées/utilisés dans les simulations

Nom d'outil	Utilité
OLSR_Scenario.tcl	Création de scénarios
Script_auto.sh	Script de configuration de scénarios

ScriptSimTrace.sh	Analyse des fichiers de traces
GnuPlot_auto.sh	Génération de figure avec GnuPlot

4.4 Résultat et analyse

Attaque 1 : Vote pour un nœud faible comme CH

La figure 4.3 montre qu'un compromis entre le nombre des nœuds moniteurs (énergie consommée) et le pourcentage de détection (niveau de sécurité). Dans cette figure qui représente l'attaque de vote pour un CH faible illégitime, tous les nœuds malicieux votent pour un nœud faible comme CH. Le pourcentage de détection des nœuds malicieux varie entre 60% à 80%. On remarque qu'avec seulement 20% des nœuds choisis comme moniteur, 60% des nœuds malicieux sont détectés. Par contre, dans le cas classique (mécanisme *Watchdog* avec 100% des nœuds choisis comme moniteurs), le pourcentage est seulement de 80 % avec nombre total des nœuds égal à 60 nœuds.

De ce qui précède, nous constatons qu'avec 50% seulement des nœuds choisis comme moniteurs peuvent atteindre la même probabilité de détection que dans le mécanisme *Watchdog*. Les ressources peuvent être optimisées dans ce cas et la durée de vie peut être prolongée.

Les figures suivantes peuvent être lues de façon similaire. Elles représentent la probabilité que les nœuds moniteurs détectent les nœuds malicieux pour les attaques citées dans le chapitre précédent. Nous remarquons qu'au fur et à mesure que le réseau devient dense, la probabilité de détection des nœuds malicieux augmente. Ce résultat s'explique par le fait que dans un réseau dense il y aura plus de chance d'avoir un nœud moniteur près d'un nœud malicieux et donc plus de chance de détecter un comportement malicieux.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Nous remarquerons pour toutes les figures qui suivent que, en fonction de la complexité de l'attaque, de la sévérité de l'attaque et des conditions de détection la probabilité de détection peut varier et un compromis entre la sécurité et la durée de vie devient possible ou non.

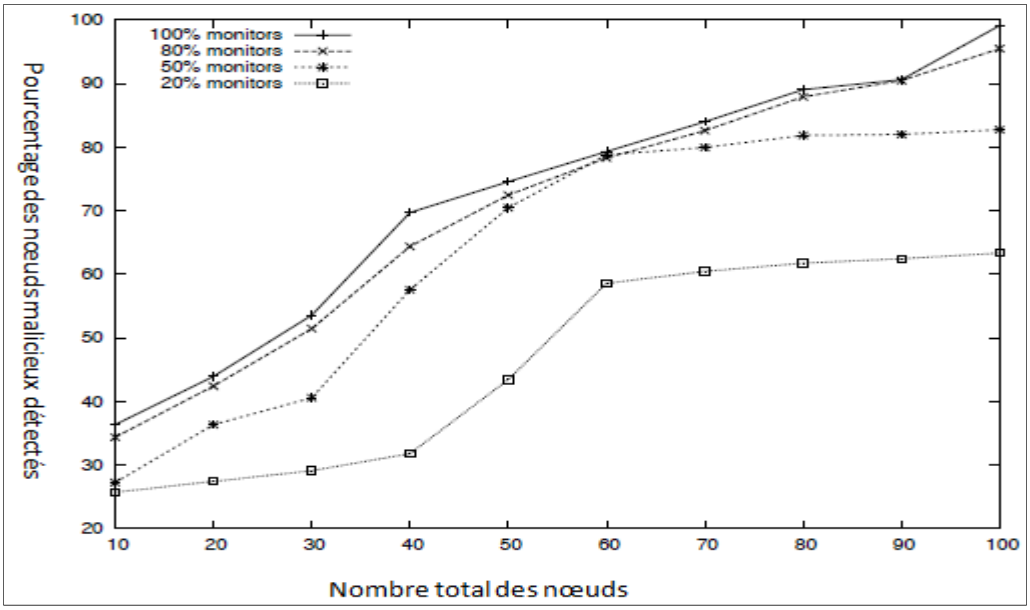

Figure 4.3: Pourcentage de détection des nœuds malicieux pour l'attaque vote pour le noeud faible comme CH

Attaque 2 : Vote pour plusieurs CH

La figure 4.4 (attaque du vote pour plusieurs CH) montre des résultats plus importants que ceux présentés dans la figure précédente. En effet, avec seulement 40% des nœuds moniteurs le système détecte entre 70% à 80 % des nœuds malicieux. Ce qui implique que le système proposé peut remplacer le mécanisme *Watchdog* et optimiser les ressources.

Attaque 3 : Suppression d'un voisin du message HELLO

La figure 4.5 montre qu'un compromis entre le nombre de nœuds (énergie consommée) et la probabilité de détection n'est pas évident. En effet, contrairement aux attaques précédentes le pourcentage des nœuds malicieux détectés est faible quand 20 % des nœuds sont engagés

pour la détection. Heureusement, l'attaque suppression d'un voisin du message HELLO n'a pas un grand impact sur le protocole SC-OLSR surtout dans un réseau dense où un nœud a plusieurs autres voisins dans ce cas.

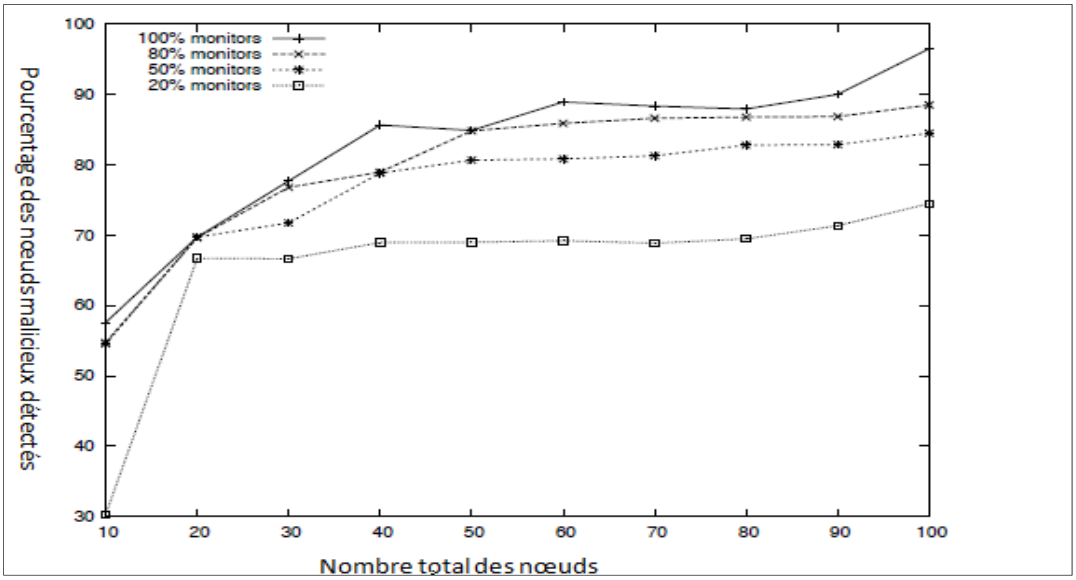

Figure 4.4: Pourcentage de détection des nœuds malicieux contre l'attaque vote pour des CH multiples

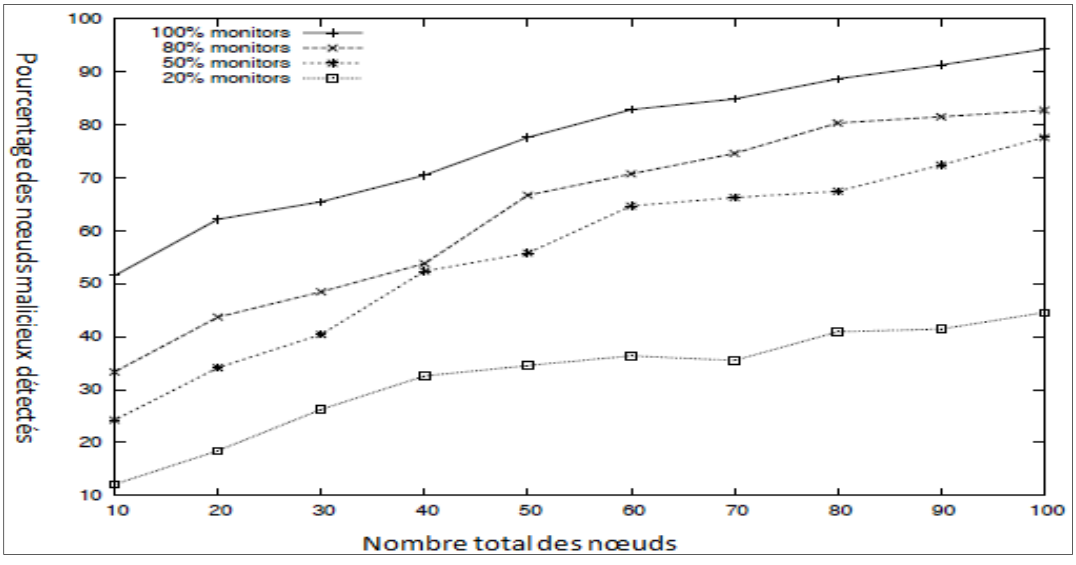

Figure 4.5: Pourcentage des nœuds malicieux détectés pour l'attaque suppression d'un voisin dans le message HELLO

Attaques 4 et 5 : Suppression/ajout d'un nœud électeur/sélectionneur dans le message TC

Dans les figures 4.6 et 4.7, les nœuds malicieux sont détectés de façon similaire en utilisant les critères les plus généraux avec des conditions moins contraignantes, ce qui explique le pourcentage de détection remarqué.

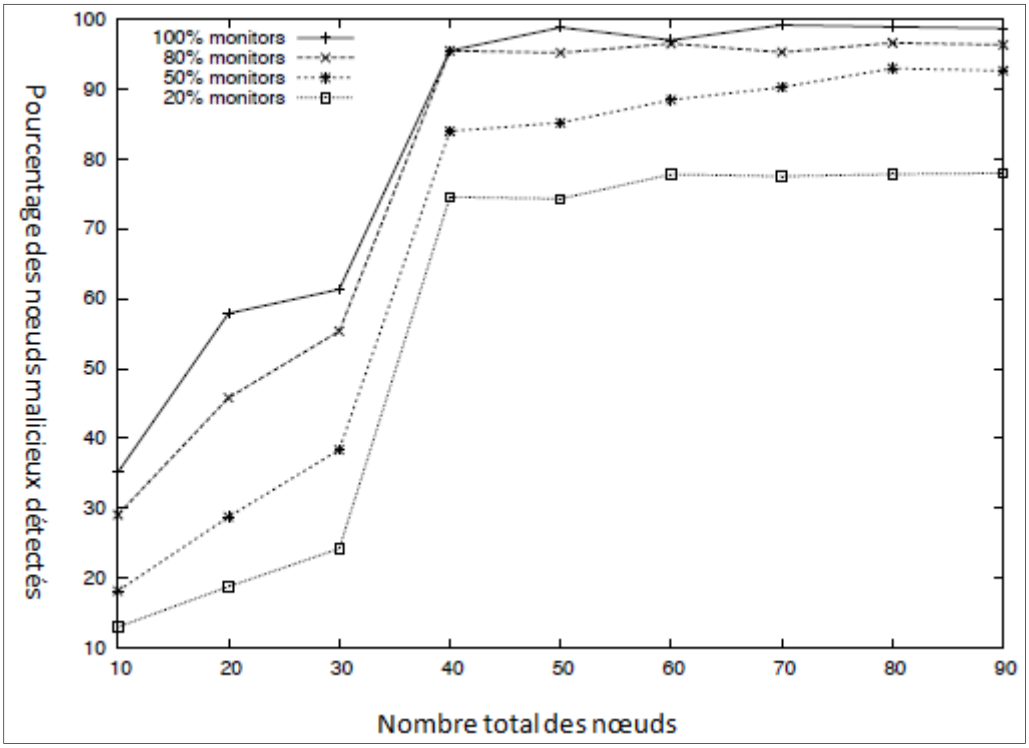

Figure 4.6 : Pourcentage des nœuds détectés pour l'attaque suppression d'un sélectionneur légitime

Les travaux expérimentaux menés ont confirmé les pistes de recherches choisies. En effet les ressources peuvent être optimisées en diminuant le nombre des nœuds de surveillance. Cette optimisation est concrétisée par la répartition aléatoire des nœuds de surveillance IDS ce qui garantit une politique de surveillance et détection imprévisible.

Les résultats ont été obtenus dans des conditions expérimentales contraignantes, ce qui nous laisse penser qu'ils peuvent être améliorés davantage si on utilise des conditions moins contraignantes comme la mobilité ou un faible pourcentage de nœuds malicieux.

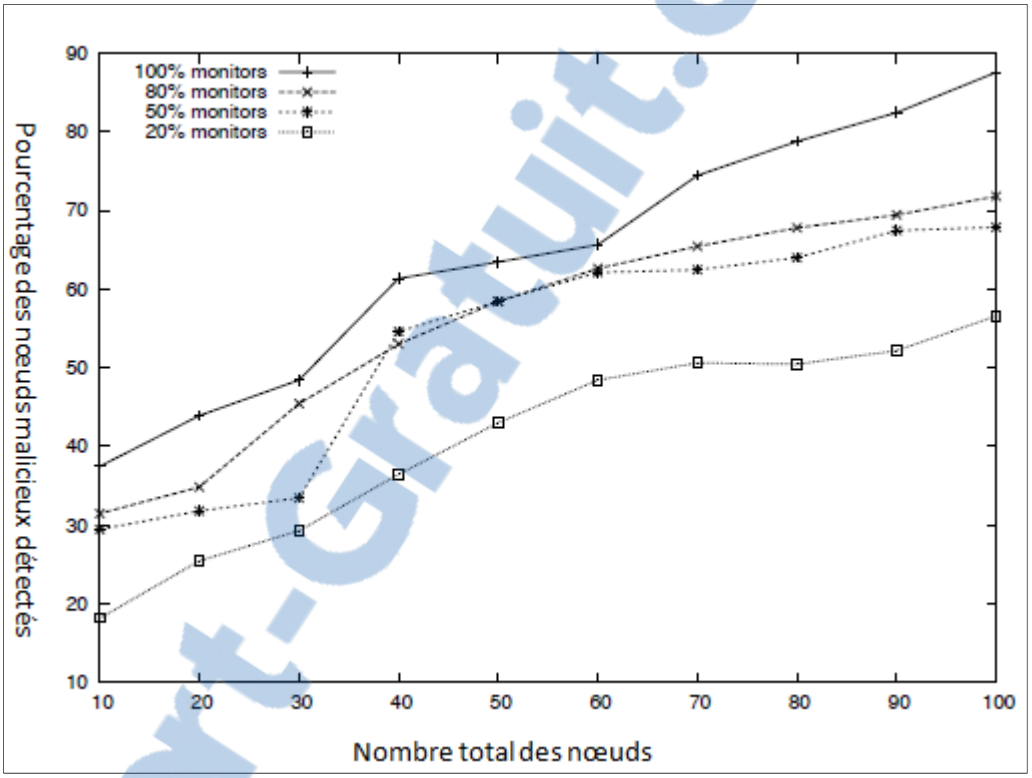

Figure 4.7 : Pourcentage des nœuds détectés pour l'attaque ajout d'un sélectionneur illégitime

Rapport-Gratuit.com

CONCLUSION

Les réseaux Ad-Hoc sont considérés comme une technologie à grande valeur ajoutée, par leur autonomie, leur flexibilité, leur auto-configuration ainsi que le fait d'offrir un système de communication sans exiger une infrastructure dédiée. Toutefois l'absence de spécifications claires de sécurité constitue un grand défi pour le déploiement de telle technologie à grande échelle d'où l'importance de concevoir un système de sécurité capable de protéger ce type de réseaux.

Durant ce travail de recherche, nous nous sommes intéressés à proposer un mécanisme de sécurité pour une extension du protocole OLSR nommée SC-OLSR. Cette extension présente l'avantage d'optimiser les ressources consommées dans le réseau en adoptant d'une part une organisation des nœuds en grappes et un modèle d'élection du chef de grappe basé sur l'énergie résiduelle.

Nos travaux ont débuté par l'analyse des vulnérabilités du protocole SC-OLSR. À l'issue de laquelle nous avons établi un modèle d'attaque afin de caractériser les adversaires et connaitre les différents modes d'attaque. Nous avons cerné notre étude aux attaques visant le processus de construction de la table de routage vu son importance mais surtout son impact sur le fonctionnement du protocole.

Afin de pouvoir détecter les nœuds malicieux, nous avons conçu un système de détection d'intrusion basé sur les spécifications protocolaires de SC-OLSR. Chaque nœud IDS peut détecter un nœud malicieux en analysant les paquets émis par ce dernier et les comparer aux spécifications. En fonction de chaque type de paquet et du rôle occupé par le nœud malicieux dans une grappe (électeur, MPR ou chef de grappe), le nœud moniteur performe l'analyse adéquate pour extraire du paquet les informations nécessaires pour la détection. Chaque nœud IDS possède désormais l'habilité de 'raisonner' afin qualifier un nœud de malicieux ou non à partir de son comportement.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Le système IDS conçu est composé de plusieurs critères de détection qui caractérise un comportement ‘normal’ d’un nœud. Nous avons conçu notre IDS dans l’optique d’offrir un système qui n’est pas complexe ni gourmand en ressource pour respecter l’objectif majeur du protocole SC-OLSR à savoir prolonger la durée de vie du réseau.

Pour évaluer notre IDS, nous avons implanté d’abord le protocole SC-OLSR en C++, modifiant ainsi un package de base appelé UM-OLSR. La validation des spécifications a été faite à l’aide de fichiers de scenarios simulés sous NS2 pour générer la topologie du réseau. Ensuite, l’IDS est implanté de façon modulaire afin de permettre son extension dans le futur. La validation de l’ensemble (IDS et SC-OLSR) est faite par des scripts que nous avons conçus pour simuler l’ensemble des attaques déterminées dans le modèle d’attaque que nous avons établi. Finalement, devant le nombre excessif des fichiers de traces générés par notre code (rapporter les événements de détection des nœuds malicieux), nous avons conçu et automatisé des outils d’analyse statistique de ces fichiers pour exploiter les résultats de la simulation et tracer les graphes d’analyse.

Nous avons effectué notre évaluation expérimentale dans le but de tester si l’IDS conçu détecte effectivement les attaques selon le modèle d’attaque établi. De plus l’objectif majeur des expériences conduites était de quantifier le nombre des nœuds moniteurs nécessaires pour obtenir un niveau de sécurité acceptable. Les résultats ont été comparés avec l’approche classique *Watchdog* (Marti, 2003) qui propose d’engager tout les nœuds dans l’opération de détection. Cette approche présente certes, le cas idéal pour la détection mais elle ne prend pas en compte les ressources consommées pour les opérations de détection ce qui affecte bien sûr la durée de vie du réseau.

Les résultats obtenus ont montré qu’un compromis est toujours possible entre l’optimisation de ressources (le nombre des nœuds moniteurs) et le niveau de sécurité souhaité. En effet pour la plupart des attaques, nous avons trouvé que si on utilise un nombre limité de nœuds comme moniteurs -répartis de façon aléatoire- on peut avoir un pourcentage acceptable de

nœuds malicieux détectés. De ce fait l'approche *Watchdog* pourrait être remplacée par notre système pour optimiser les ressources et prolonger la durée de vie du réseau.

Notre approche ouvre plusieurs perspectives de recherche future. En effet, la simulation des scenarios mobiles pourrait donner de meilleurs résultats. Nous estimons que si les nœuds moniteurs sont en déplacement, ils pourront détecter plus de nœuds malicieux ainsi la probabilité de détection pourrait s'améliorer. Nous pouvons aussi proposer d'autres stratégies de déploiement des nœuds moniteurs pour optimiser les ressources énergétiques et améliorer la détection des nœuds malveillants. Et finalement proposer un système de réponse pour punir les nœuds malveillant détectés.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Abdellaoui, R., et J-M. Robert. 2009. « SU-OLSR: A New Solution to Thwart Attacks against the OLSR Protocol ». In *Proceeding of conference on Security in network architectures and information systems (SAR-SSI)*.

Adjih, C., T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler et D. Raffo. 2003. « Securing the OLSR protocol ». In *Proceedings of the 2nd IFIP Med-Hoc-Net*, p. 25–27.

Adjih, C, T. Clausen, A. Laouiti, P. Muhlethaler et D. Raffo. 2005a. « Securing the OLSR routing protocol with or without compromised nodes in the network ». *INRI Research Report RR-5494,* 55 pages.

Adjih, C, D. Raffo et P. Muhlethaler. 2005b. « Attacks Against OLSR: Distributed Key. Management for Security ». In *Proceeding of 2005 OLSR Interop and Workshop,* Ecole Polytechnique, France.

Adnane, A., C. Bidan et R. T. de Sousa Jr. 2008. « Validation of the OLSR routing table based on trust reasoning ». In *Proceedings of the International Workshop on Trust in Mobile Environments.*

Beijar, N. 2002. « Zone routing protocol (ZRP) ». Ad Hoc Networking, Licentiate course on Telecommunications Technology.

Beyer, D. A. 1990. « Accomplishments of the DARPA SURAN Program ». In *Proceeding of Military Communications Conference, (MILCOM),*. p. 855-862 vol.2.

Biradar, Rajashree V., et V. C. Patil. 2006. « Classification and Comparison of Routing Techniques in Wireless Ad Hoc Networks ». In *Proceeding of International Symposium on Ad Hoc and Ubiquitous Computing (ISAUHC),* p. 7-12.

Buchegger, S., et J-Y. Le Boudec. 2002. « Performance analysis of the CONFIDANT protocol: Cooperation of nodes—fairness in dynamic ad-hoc networks ». In *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, p. 226–236.

Cheng. H.T, H. Shan et W. Zhuang. 2010. « Infotainment and road safety service support in vehicular networking: From a communication perspective ». In *Mechanical Systems and Signal Processing*, vol. 25, no 6, p. 2020-2038.

Chriqi, A., H. Otrok et J-M. Robert. 2009. « SC-OLSR: Secure Clustering-Based OLSR Model for Ad Hoc Networks ». In *Proceeding of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB),* p. 239-245.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Clausen, T., et P. Jacquet. 2003. « RFC3626 : Optimized Link State Routing Protocol (OLSR) ».

Fall, .K, et K. Varadhan. 2008. « The ns Manual ». The VINT Project, UC Berkeley, LB USC/ISI, and Xerox PARC. En ligne <www.isi.edu/nsnam/ns/doc/>.

Fourati, A., et K.A. Agha. 2006. « A shared secret-based algorithm for securing the OLSR routing protocol ». *Telecommunication Systems*, vol. 31, no 2, p. 213-226.

Hass, Z.J., M.R. Pearlman et P. Samar. 2002. « The Zone Routing Protocol (ZRP) for Ad Hoc Networks ». *draft-ietf-manet-zone-zrp-04.txt, IETF MANET, Internet Draft.*

Hartenstein, H., et K. P. Laberteaux. 2008. « A tutorial survey on vehicular ad hoc networks ». *IEEE Communications Magazine*, vol. 46, no 6, p. 164-171.

Janert, P. 2008. *Gnuplot in Action,* EARLY Access Edition. En ligne. <http://www.gnuplot.info/>.

Johnson, D.B., D.A. Maltz et J. Broch. 2001. « DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks ». *Ad hoc networking*, p. 139–172, Addison-Wesley.

Kahn, R. E., S. A. Gronemeyer, J. Burchfiel et R. C. Kunzelman. 1978. « Advances in packet ratio technology ». *Proceedings of the IEEE*, vol. 66, no 11, p. 1468-1496.

Labovitz, C., A. Ahuja, A. Bose et F. Jahanian. 2001. « Delayed Internet routing convergence ». *IEEE/ACM Transactions on Networking,* vol. 9, no 3, p. 293-306.

Marti, S., T.J. Giuli, K. Lai et M. Baker. 2000. « Mitigating routing misbehavior in mobile ad hoc networks ». In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, p. 255-265

McQuillan, J., I. Richer et E. Rosen. 1980. « The New Routing Algorithm for the ARPANET». *IEEE Transactions on Communications,* vol. 28, no 5, p. 711-719.

Michiardi, P., et R. Molva. 2002. « Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks ». In *Proceedings of the Sixth IFIP conference on security communications, and multimedia*.

Mishra, A., K. Nadkarni et A. Patcha. 2004. « Intrusion detection in wireless ad hoc networks». *IEEE Wireless Communications,* vol. 11, no 1, p. 48-60.

Mohammed, N., H. Otrok, Wang Lingyu, M. Debbabi et P. Bhattacharya. 2008. « A Mechanism Design-Based Multi-Leader Election Scheme for Intrusion Detection in MANET ». In *Proceeding of Wireless Communications and Networking Conference,* p. 2816-2821.

Perkins, C. E., et P. Bhagwat. 1994. « Highly dynamic Destination-Sequenced Distance Vector routing (DSDV) for mobile computers ». *ACM SIGCOMM Computer Communication Review,* vol. 24, no 4, p. 234-244.

Perkins, C. E., E. M. Royer et S. Das. 2002. « Ad-hoc On-demand Distance Vector (AODV) Routing ». *draft-ietf-manet-aodv-10.txt, IETF MANET, Internet Draft.*

Raffo, D., C. Adjih, T. Clausen et P. Mühlethaler. 2004. « An advanced signature system for OLSR ». In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, p. 10-16.

Roy, J. F. 2006. *UM-OLSR*. University of Murcia: MASIMUM. En ligne. <http://masimum.dif.um.es/?Software:UM-OLSR>.

Wang, M., L. Lamont, P. Mason et M. Gorlatova. 2005. « An effective intrusion detection approach for OLSR MANET protocol ». In *Proceedings of the 1st IEEE ICNP Workshop on Secure Network Protocols (NPSec*), p. 55-60.

Yick, J., B. Mukherjee et D. Ghosal. 2008. « Wireless sensor network survey ». *Computer Networks*, vol. 52, no 12, p. 2292-2330.

Zhang, et D. Y. C. Kiat. 2010. « A Novel Architecture of Intrusion Detection System ». In *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC),* p. 1-5.

Wireless Local Area Networks. En ligne <http://www.ieee802.org/11>.