REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB BLIDA

Faculté des sciences

Département : Informatique

Mémoire

Pour l'obtention du diplôme de

Master

En informatique

Option : Sécurité des Systèmes d'Information (SSI)

Présenté par :

Boukhebiza Meroua

THEME

La sécurité du protocole de routage OLSR dans un réseau AD HOC mobile

Encadré par : Mme Bouaissa Djamila.

Soutenu le : 20/10/2020

Devant le jury composé de :

Mme Boustia Narhimene … Présidente

Mme Arkam Meriem … Examinatrice

Année Universitaire 2019/2020

Remerciements

*Nous remercions **Dieu** de nous avoir accordé des connaissances de la science et de nous avoir aidés à réaliser ce travail.*

Si ce mémoire a pu voir le jour, c'est certainement grâce au soutien et l'aide de plusieurs personnes qui nous ont permis d'accomplir ce travail dans des conditions idéales. On profite de cet espace pour les remercier tous.

Au terme de ce modeste travail nous tenons à remercier chaleureusement et respectivement tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste projet de fin d'étude.

*Nos vifs remerciements vont tous d'abord à mon encadreur **Mme. BOUAISSA DJAMILA** qui m'a encadré tout le long de ce projet.*

Tout notre respect et nos remerciements vont vers nos jurys qui ont pleinement consacrer leur temps et leur attention afin d'évaluer notre travail, qui espérons le sera à la hauteur de leur attente.

Enfin, nos remerciements les plus sincères sont adressés à tous les professeurs et l'administration.

Dédicace

À l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi mon père.

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur ; maman que j'adore.

Aux personnes dont j'ai souhaitais leurs présence dans ce jour, à mes frères Anis et Mohamed amine, mon fiancé à tous ceux que j'aime, à tous ceux qui m'aiment.

je dédie ce travail aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études, mes aimables amis, collègues d'étude, et collègues de travail.

Boukhebiza Maroua

Sommaire

Liste des figures

Liste des tables

Résumé

Les protocoles de routage ad hoc existant dans la littérature font l'hypothèse d'un environnement idéal dans lequel le fonctionnement du réseau n'est pas soumis à des attaques malveillantes. C'est la raison pour laquelle de nombreuses vulnérabilités au niveau du routage sont apparus. Concevoir des mécanismes infaillibles de sécurité pour les réseaux ad hoc est un challenge. D'autant plus que ces réseaux représentent des caractéristiques contraignantes et sont très vulnérables aux attaques comparées aux réseaux filaires ou les réseaux sans fil basés sur une infrastructure. Différentes techniques ont été proposées pour sécuriser certains des protocoles de routage envisagés pour les réseaux ad hoc. Ce domaine reste très complexe et encore nouveau.

Dans ce travail de recherche, nous considérons le problème de la sécurisation des informations de routage du protocole proactif OLSR. Suite à l'étude effectuée sur le protocole OLSR, ainsi que les approches de sécurité proposées pour le sécuriser, nous proposons un schéma de routage sécurisé pour OLSR. Notre approche consiste à intégrer dans le protocole OLSR des mécanismes de sécurité qui tiennent compte des ressources limités des noeuds. Un mécanisme de contre mesure sera également proposé pour contrer l'attaque du trou noir. Afin d'évaluer les performances de notre protocole, nous avons effectué plusieurs tests de simulation à travers le simulateur NS2 (Network Simulator). Les résultats obtenus montrent que notre protocole réalise un équilibre entre les performances du protocole et le niveau de sécurité offert.

Mots-clés : réseau mobile Ad hoc, OLSR, routage sécurisé, attaque du trou de noir, attaque du trou de ver, simulation, NS2.

Abstract

Currently most mature Ad hoc protocols have been designed with the assumption that no misbehaving entity is present in the network. There are however many applications in which such an assumption is not acceptable and there are many ways by which it is possible to harm the network operation. Many solutions have been performed to secure Ad hoc protocols. However, the security of MANETs still very complex.

In this work, we consider the problem of securing routing information of the proactive protocol OLSR. We propose a secure routing scheme for OLSR. Our approach consists to integrate in OLSR security mechanisms that take into account the limited resources of nodes. We propose also a counter-measure to reinforce the security of OLSR against the black hole attack. We use the ns2 simulator to test the performance of our solution. The obtained results show that our protocol presents a compromise between robustness, in terms of security, and protocol performance.

Keywords: mobile ad hoc networks, OLSR, secure routing, black hole attack, worm hole attack, simulation, NS2.

INTRODUCTION GÉNÉRALE

Aujourd'hui, les réseaux sans fil ont connu une forte expansion et sont de plus en plus populaires du fait de leur facilité de déploiement. L'évolution rapide de la technologie dans le domaine de la communication sans fil, a permis aux usagers munis d'unités de calcul portables d'accéder à l'information à n'importe quel moment depuis n'importe quel endroit. Cet environnement n'astreint plus l'usager à une localisation fixe, mais lui permet une libre mobilité tout en assurant sa connexion avec le réseau. Il offre des solutions ouvertes pour fournir des services essentiels là où l'installation d'infrastructures n'est pas possible.

Les réseaux sans fil sont généralement classés selon deux catégories : les réseaux sans fil avec infrastructure fixe qui utilisent généralement le modèle de la communication cellulaire et les réseaux sans fil sans infrastructure fixe, appelés aussi réseaux Ad hoc. Un réseau Ad hoc est un ensemble autonome et coopératif de nœuds mobiles qui se déplacent et communiquent par une transmission sans fil qui ne suppose pas d'infrastructure préexistante. Le réseau Ad hoc se forme de manière spontanée et provisoire dès que plusieurs nœuds mobiles se trouvent à portée radio les uns des autres. Les nœuds communiquent, selon la distance qui les sépare, par deux modes de communication : soit les nœuds mobiles peuvent directement communiquer (en transmission Ad hoc) car ils sont à portée de transmission, soit ils doivent utiliser d'autres nœuds mobiles comme des relais pour acheminer les paquets à destination.[3]

Le sujet de notre mémoire entre dans le cadre de l'étude du problème de la sécurité du routage dans les réseaux mobiles ad hoc. Notre étude offre principalement une étude synthétique des travaux de recherche qui ont été faits, et qui se font à l'heure actuelle, dans le but de résoudre le problème de sécurité d'acheminement de données de contrôle entre les hôtes mobiles du réseau ad hoc. Comme nous allons le voir, le problème de routage est très compliqué, cela est dû essentiellement à la propriété des réseaux ad hoc qui se caractérisent par une double absence : celle d'une infrastructure fixe celle de toute administration centralisée.

Dans notre travail, nous traitons la sécurité au niveau de la couche réseau tout en essayant de travailler sur un type de protocole de routage dans les réseaux ad hoc (OLSR) en vue de généraliser la solution pour d'autres protocoles. Ce mémoire se divise en quatre chapitres : dans le premier chapitre nous présentons les réseaux mobiles Ad hoc et les principaux concepts liés à ces environnements. Dans le deuxième chapitre, nous introduisons le problème de sécurité du routage dans cet environnement et dans le troisième chapitre, nous parlons sur le problème de sécurité du protocole de routage **OLSR**.

Le dernier chapitre est consacré à la simulation des réseaux Ad hoc par l'utilisation de protocole de routage **OLSR** et l'attaque **BLACKHOLE** a fin d'étudié la solution proposée pour la sécurité de ce dernier et étudier aussi l'influence de l'attaque sur notre protocole étudié.

Chapitre 01

Généralités sur les réseaux Ad hoc

1.1. Introduction

La naissance de la technologie sans fil offre aujourd'hui de nouvelles perspectives dans le domaine des télécommunications. L'évolution récente des moyens de la communication sans fil a permis la manipulation de l'information à travers des unités (sites) portables qui ont des caractéristiques particulières et accèdent au réseau à travers une interface de communication sans fil. C'est l'environnement mobile, qui permet aux sites, une libre mobilité et il ne pose aucune restriction sur la localisation des usagers.

Les environnements mobiles offrent une grande flexibilité d'emploi. En particulier, ils permettent la mise en réseau des sites dont le câblage serait trop onéreux à réaliser dans sa totalité**.** Les réseaux sans fil Ad Hoc ne nécessitent aucune infrastructure préalable. L'avantage de cette topologie est qu'elle permet de déployer un réseau dans un délai très court avec un coût réduit et d'une manière spontanée. L'évolution récente de la technologie dans le domaine de la communication sans fil et l'apparition des unités de calcul portables poussent aujourd'hui les chercheurs à faire des efforts afin de réaliser le but des réseaux **: [2]**

« L'accès à l'information n'importe où et n'importe quand ».

Dans ce chapitre nous allons présenter les environnements mobiles et les principaux concepts liés à ces environnements. Nous commençons par la définition de cet environnement et les deux classes qui le constituent (mode infrastructure et mode sans infrastructure). Nous introduisons ensuite le concept des réseaux Ad hoc et les caractéristiques inhérentes à ces réseaux, après nous définissons quelques domaines d'application d'un réseau Ad hoc, et enfin on va examiner la notion du routage sur les réseaux Ad hoc.

1.2. Historique

À l'origine les réseaux Ad hoc sont utilisés pour les applications militaires (réseau tactique) pour améliorer et garantir la communication dans les champs de bataille, l'absence d'une infrastructure est recommandée dans ce genre d'environnement.

Au début des années 70, la première utilisation d'un réseau avec un support radio au sein de projet **«** Packet Radio Network **»** **PRNet** en 1973 de **DARPA «** The Defense Advanced Research Projects Agency **»**, il dispose d'une architecture distribuée qui partage le canal de diffusion **«** broadcast **»** en utilisant une combinaison des méthodes **Aloha** et **CSMA** pour l'accès au canal avec une technique de routage store-and-forward multi-hop qui élargir la zone de couverture par répétition des paquets.

Par la suite, en 1983, le **«** Survivable Radio Networks **»** **SURAN** a été développé aussi par **DARPA**. L'objectif était d'étendre le réseau afin de dépasser les limitations **«** en particulier permettre le passage à des réseaux comportant énormément de nœuds, gérant la sécurité, l'énergie… **»**. En 1987, l'introduction des technologies **LPR«** Low-cost Packet Radio**»** et **SCN«** Survivable Communication Network**»**, et plusieurs d'autres projets qui portent sur ce domaine tel que : **GloMo «** Global Mobile**»** qui fournis des services multimédias sur une connexion sans fil, **WINGs «** Wireless Internet**»** Gateway une architecture réseau Peer-to-Peer , **MMWN «** Multimedia Mobile Wireless Network de GTE Internet working**»** une architecture réseau à base des clusters, **TI «** Tactical Internet**»** une implémentation d'un réseau mobile multi saut en 1997, **ELB ACTD «** Extending the Littoral Battle-space Advanced Concept Technology Demonstration **»** un autre réseau Ad hoc financé par l'armée américaine en 1999.

Les recherches sont apparues dans le monde commercial au années 90 avec l'apparition de protocole 802.11 de **l'IEEE** (Institute of Electrical and Electronics Engineers). Le groupe de travail **MANET «** Mobile Ad hoc Network **»** de **l'IETF «** Internet Engineering Task Force **»** est l'un des groupes actifs qui s'intéressent aux réseaux Ad hoc**.[3]**

.

1.3. Les réseaux sans fil

1.3.1. Définition d'un réseau sans fil

Un réseau sans fil en anglais **« wireless network »** est un réseau dans lequel les différents éléments participants (ordinateur portable, téléphone portable…etc.) ne sont pas raccordés entre eux par un média physique. La transmission des données se fait via les ondes hertziennes (radio ou infrarouge). Ceci permet aux utilisateurs de se déplacer dans un périmètre de couverture pouvant aller d'une dizaine de mètres à quelques kilomètres**.[4]**

1.3.2. Les architectures d'un réseau sans fil

1.3.2.1. Mode avec infrastructure

Ce mode de fonctionnement est très semblable au protocole Ethernet des réseaux filaires. Les machines se connectent à un point d'accès appelé aussi station de base, qui partage la bande passante disponible. Les stations de base sont munies d'une interface de communication sans fil avec les sites mobiles qui se trouvent dans sa zone géographique ou sa couverture radio. Ce mode de fonctionnement est illustré à la figure suivante : **[2]**

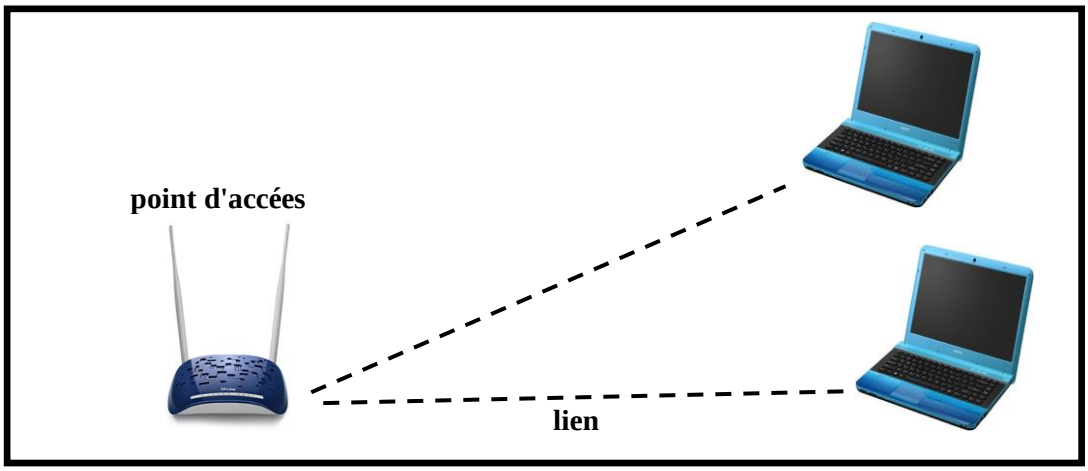


Figure 1 : Mode avec infrastructure.

1.3.2.2. Mode sans infrastructure ou réseau Ad hoc

Ce mode n'a pas besoin de point d'accès pour fonctionner, ce sont les stations elles- mêmes qui entrent en communication sans s'appuyer sur un équipement extérieur. Tous les nœuds d'un réseau de ce type se comportent comme des routeurs et prennent part à la découverte et à la maintenance des chemins de communication entre les différentes machines. Ce type de réseau s'organise lui-même.

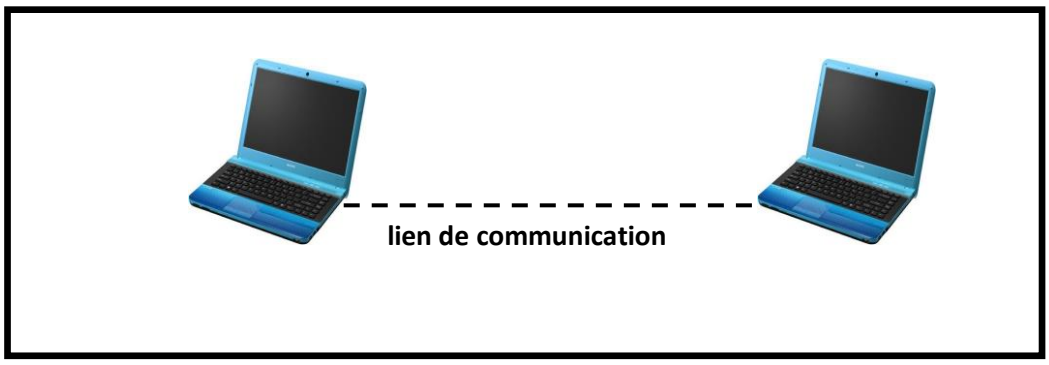


Figure 2 : Mode sans infrastructure ou Ad hoc.

1.4. Les réseaux mobiles Ad hoc

1.4.1. Définition

Un réseau Ad Hoc appelé généralement MANET **« Mobile Ad hoc Network »**, est composé d'un ensemble relativement danse de nœuds mobiles qui se déplacent librement dans une certaine zone géographique sans aucune infrastructure fixe préexistante. Un nœud dans le réseau Ad hoc communique avec un autre nœud directement (en utilisant son interface sans fil), si ce dernier est dans sa portée de transmission, ou indirectement par l'intermédiaire d'autres nœuds du réseau dans le cas contraire. Chaque nœud dans le réseau Ad hoc doit se comporter comme un terminal, et aussi comme un routeur, et participer à la découverte et la maintenance des routes entre les nœuds du réseau. Il y a aucune limitation de taille dans un réseau Ad hoc, il peut contenir des dizaines ou des milliers de nœuds**.[5]**

1.4.2. Les caractéristiques des réseaux Ad hoc

Les réseaux mobiles Ad hoc sont caractérisés par ce qui suit :

- **Une topologie dynamique :** Les unités mobiles du réseau se déplacent d'une façon libre et arbitraire. Par conséquent la topologie du réseau peut changer à des instants imprévisibles, d'une manière rapide et aléatoire. Les liens de la topologie peuvent être unis ou bidirectionnels.

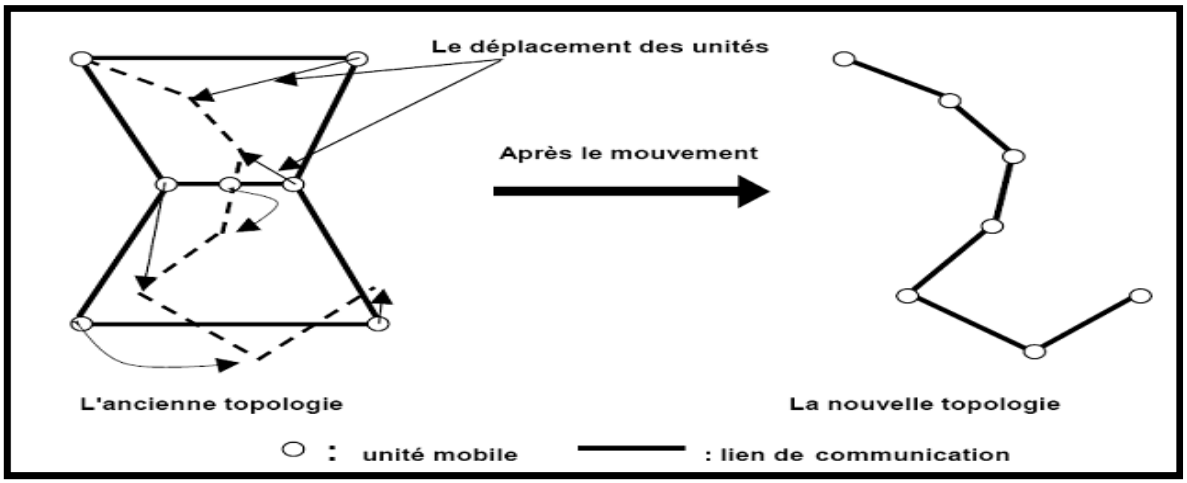


Figure 3 : Le changement de la topologie. [7]

- **L'absence d'infrastructure :** Les réseaux Ad hoc se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructures préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue**.[7]**

- **Interférences :** Dans un réseau Ad hoc, les liens radio ne sont pas isolés. Ceci peut impliquer que deux transmissions simultanées sur une même fréquence ou sur des fréquences proches peuvent interférer et provoquer des erreurs de transmission. Un grand nombre de paquets peuvent être endommagés et perdus lors du transfert.

- **Multihops ou multi saut :** Un réseau Ad hoc est qualifié par « **Multihops** », car plusieurs nœuds mobiles peuvent participer au routage et servent comme routeurs intermédiaires**. [3]**

- **Erreur de transmission :** Les erreurs de transmission radio sont plus fréquentes que dans les réseaux filaires.

- **Nœuds cachés :** Ce phénomène est très particulier à l'environnement sans fil.**[9]**
  Un exemple est illustré par la Figure 6. Dans cet exemple, les nœuds B et C ne s'entendent pas, à cause d'un obstacle qui empêche la propagation des ondes.
  Les mécanismes d'accès au canal vont permettre alors à ces nœuds de commencer leurs émissions simultanément. Ce qui provoque des collisions au niveau du nœud A.

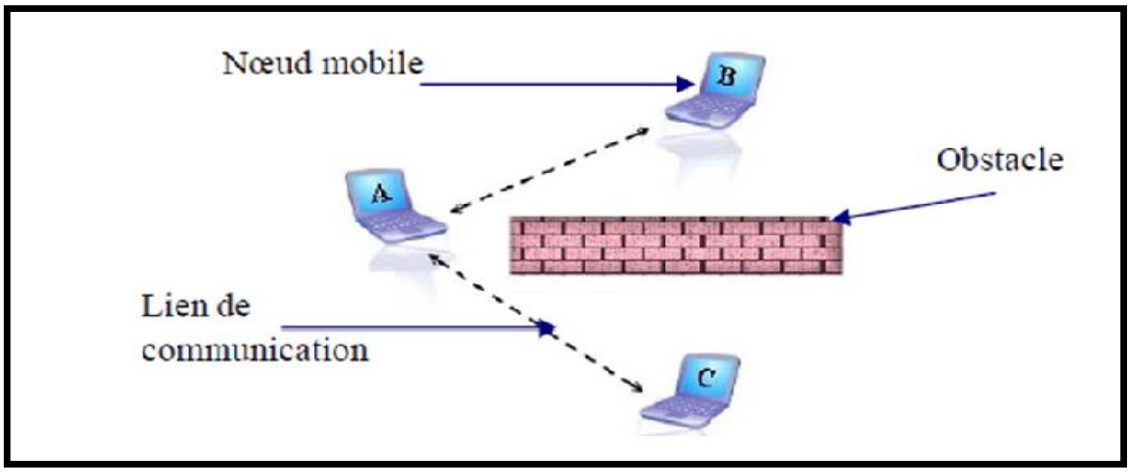


Figure 4 : nœuds cachés. [9]

1.4.3. Domaines d'utilisation des réseaux Ad hoc

Les réseaux Ad hoc sont utilisés dans toutes les applications où le déploiement d'une architecture centralisée est contraignant. En effet, la robustesse, le coût réduit et le déploiement rapide qu'ils présentent leur confèrent un accès à une large palette d'applications dont **:**

- **Les applications militaires :** Les réseaux Ad hoc ont été utilisés la première fois par l'armée. En effet ce type de réseaux est la solution idéale pour maintenir une communication sur un champ de bataille entre les différentes troupes unités d'une armée.

- **Les opérations de secours :** Dans les zones touchées par les catastrophes naturelles (cyclone, séisme, etc.), le déploiement d'un réseau Ad hoc est indispensable pour permettre aux unités de secours de communiquer.

- **L'utilisation à des fins éducatives :** Le déploiement d'un réseau Ad hoc lors d'une conférence ou d'une séance de cours est très judicieux car cela permet aux chercheurs et étudiants de partager des ressources (fichiers, accès à internet…etc.) et de communiquer sans avoir besoin d'une quelconque infrastructure.

- **Applications industrielles :** Des scénarios plus complexes dans le domaine industriel appelés réseaux de capteurs (Sensor Networks) peuvent former un MANET pour s'adapter à différents environnements. Un exemple d'une telle application est la formation d'un MANET pour la surveillance médicale, la détection des Feux de forêt, la surveillance des volcans…etc.

- **Mise en œuvre des réseaux véhiculaires :** sur un réseau routier les véhicules peuvent avoir besoin de communiquer entre eux ou avec leur environnement afin de partager des informations dans le but de gérer et réguler le trafic routier. Les réseaux Ad hoc sont alors la solution idéale**. [2]**

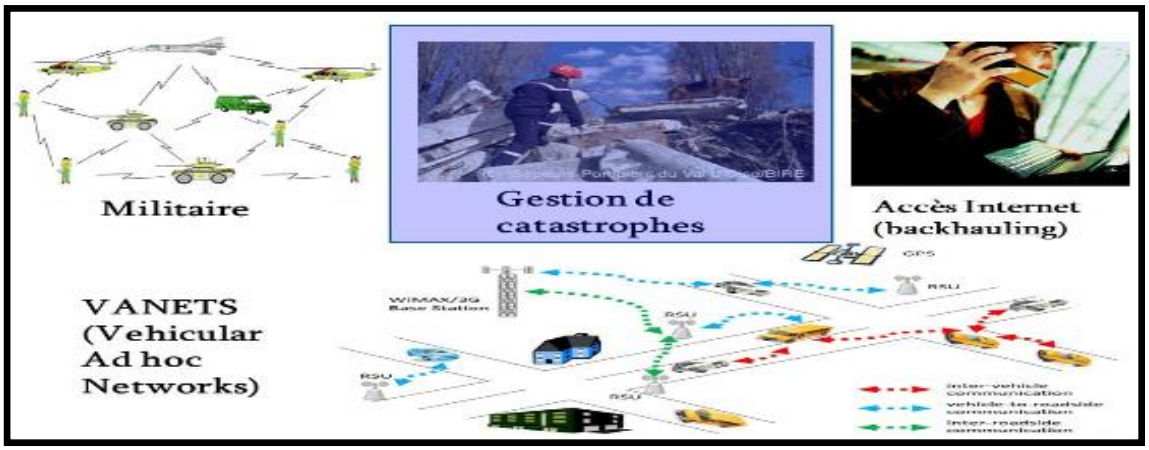


Figure 5 : Domaines d'application d'un réseau Ad hoc. [2]

1.4.4. Les avantages et les inconvénients de réseau Ad hoc

1.4.4.1. Les avantages :

Le réseau Ad hoc possède quelques avantages supplémentaires qui sont **:**

- ✓ **Simplicité de déploiement :** ne nécessite aucun pré requis puisqu'il suffit de disposer d'un certain nombre de terminaux dans un espace pour créer un réseau Ad hoc, et rapide puisqu'il est immédiatement fonctionnel dès lors que les terminaux sont présents.

- ✓ **La souplesse d'utilisation :** est un paramètre très important puisque les seuls éléments pouvant tombés en panne sont les terminaux eux-mêmes. Autrement dit, il n'y a pas de panne "pénalisante" de manière globale (une station qui sert au routage peut être remplacée par une autre si elle tombe en panne). **[9]**

- ✓ **La mobilité :** l'absence de câblages autorise les nœuds à se déplacer l'un par rapport aux autres au cours du temps.

- ✓ **Évolutifs de réseau :** pour ajouter un nœud à un réseau Ad hoc préexistant, il suffit d'approcher le nouveau venu d'au moins de l'un des membres du réseau. De même il suffit de l'éloigner pour le retirer du réseau. **[10]**

1.4.4.2. Les inconvénients

Le réseau Ad hoc contient aussi des inconvénients tel que **:**

- ✓ **Une bande passante limitée :** une des caractéristiques primordiales des réseaux Basés sur la communication sans fil est l'utilisation d'un médium de communication Partagé. Ce partage fait que la bande passante réservée à un hôte soit modeste.

- ✓ **Des contraintes d'énergie :** les hôtes mobiles sont alimentés par des sources d'énergie autonomes donc restreintes, comme les batteries, par conséquent la durée de traitement est réduite. Donc le paramètre d'énergie doit être pris en considération dans tout contrôle fait par le système.

- **Une sécurité physique limitée :** les réseaux mobiles Ad hoc sont plus touchés par le paramètre de sécurité, que les réseaux filaires classiques. Les terminaux ne sont pas protégés, ils sont menacés de vol ou de destruction**. [8]** Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.

- ✓ **Problème de sécurité :** la sécurité dans les réseaux AD HOC est difficile à contrôler, notamment parce que dans l'interface air l'écoute clandestine est très simple à réaliser. **[11]**

1.5. Le routage dans les réseaux ad hoc

1.5.1. Définition du routage

Généralement, Le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexions défini. Son intérêt consiste à trouver le chemin optimal au sens d'un certain critère de performance (bande passante, délai, etc.). Il doit aussi être capable de s'adapter aux événements venant perturber le réseau (panne, congestion, etc.).**[13]**

Par exemple si on suppose que les coûts des liens sont identiques, le chemin indiqué dans la figure suivante est le chemin optimal reliant la station source et la station destination. Une bonne stratégie de routage utilise ce chemin dans le transfert des données entre les deux stations**.**

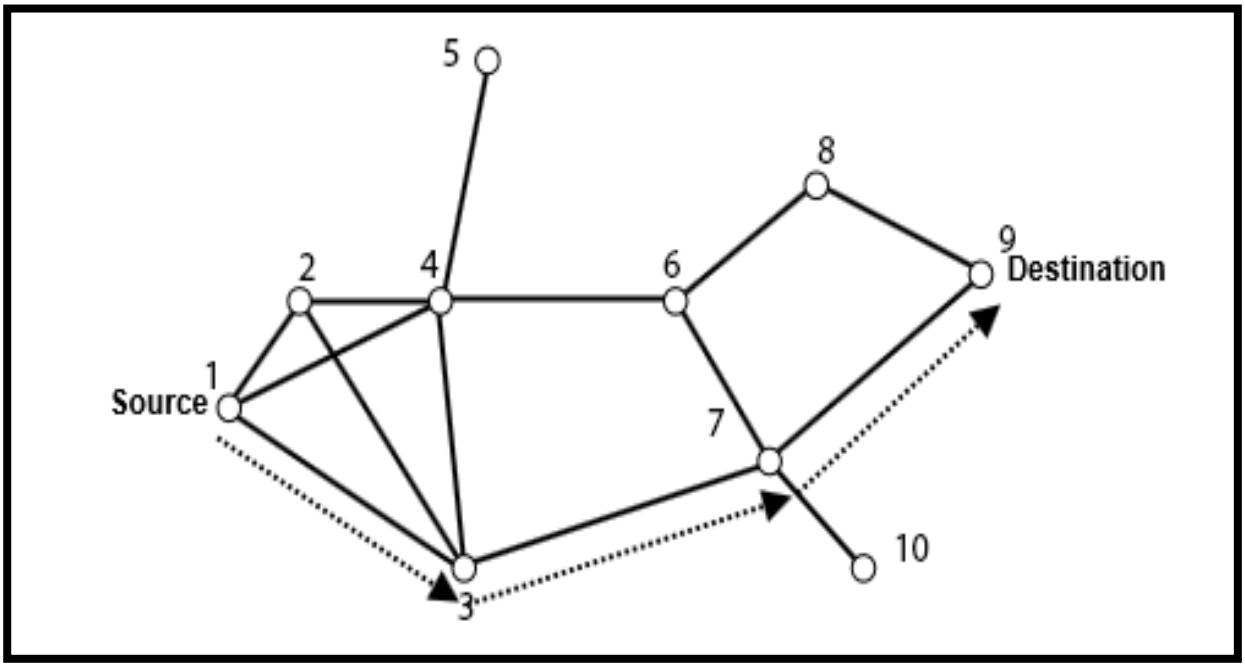


Figure 6 : Le chemin utilisé dans le routage entre la source et le destinateur. [14]

1.5.2. Classification des protocoles de routage

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en : **Proactifs**, **Réactifs** et **Hybrides.**

De nombreux protocoles et algorithmes ont été proposés pour rendre la communication dans les réseaux Ad hoc plus efficace. Et leurs performances ont été analysées dans différentes situations. Dans la section suivante nous allons présenter certains protocoles de routage du monde Ad hoc développés dans le cadre du groupe de travail MANET de l'IFTE. Ces protocoles sont représentatifs de diverse techniques et sont les plus avancés sur la voie d'une normalisation**. [2]**

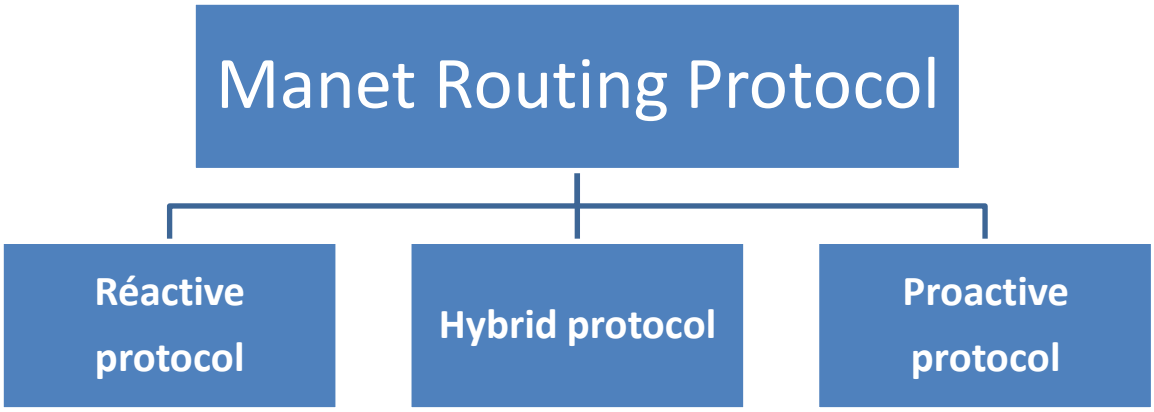


Figure 7 : Classification des protocoles de routage. [15]

1.5.2.1. Les protocoles de routage réactifs

Les protocoles de routage réactifs, dits aussi protocoles de routage à la demande, représentent les protocoles les plus récents proposés dans le but d'assurer le service du routage dans les réseaux sans fil. Les protocoles de routage appartenant à cette catégorie, créent et maintiennent les routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte globale de routes est lancée, et cela dans le but d'obtenir une information. Les protocoles de routage réactifs engendrent un trafic très important ce qui conduit souvent à la saturation rapide du réseau. Pour y remédier, les protocoles réactifs évitent au maximum les inondations qui consomment beaucoup de ressources.**[15]**

1.5.2.2. Les protocoles de routage proactifs

Les protocoles de cette catégorie sont basés sur les algorithmes classiques d'état de liens et de vecteur de distance. Les protocoles de routage proactifs essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque nœud du réseau. Les routes sont sauvegardées même si elles ne sont pas utilisées. La mise à jour permanente des tables de routage, est assurée par un échange continu des messages de mise à jour des chemins. Lorsqu'un nœud reçoit un paquet de contrôle, il met à jour ses tables de routages.

Ainsi, de nouvelles routes seront construites sur la base des informations topologiques transportées par les trames de contrôle. Ce processus est déclenché aussi à chaque changement de topologie pour reconstruire à nouveau les routes**. [2]**

1.5.2.3. Les protocoles de routage hybrides

Dans ce type de protocole, on peut garder la connaissance locale de la topologie jusqu'à une certaine distance (nombre prédéfini de sauts) par un échange périodique de trame de contrôle, autrement dit par une technique proactive. Les routes vers des nœuds plus lointains sont obtenues par schéma réactif, c'est-à-dire par l'utilisation de paquets de requête en diffusion**.** Avec ce système, on dispose immédiatement des routes dans notre voisinage proche, et lorsque la recherche doit être étendue plus loin, elle en est optimisée**. [2]**

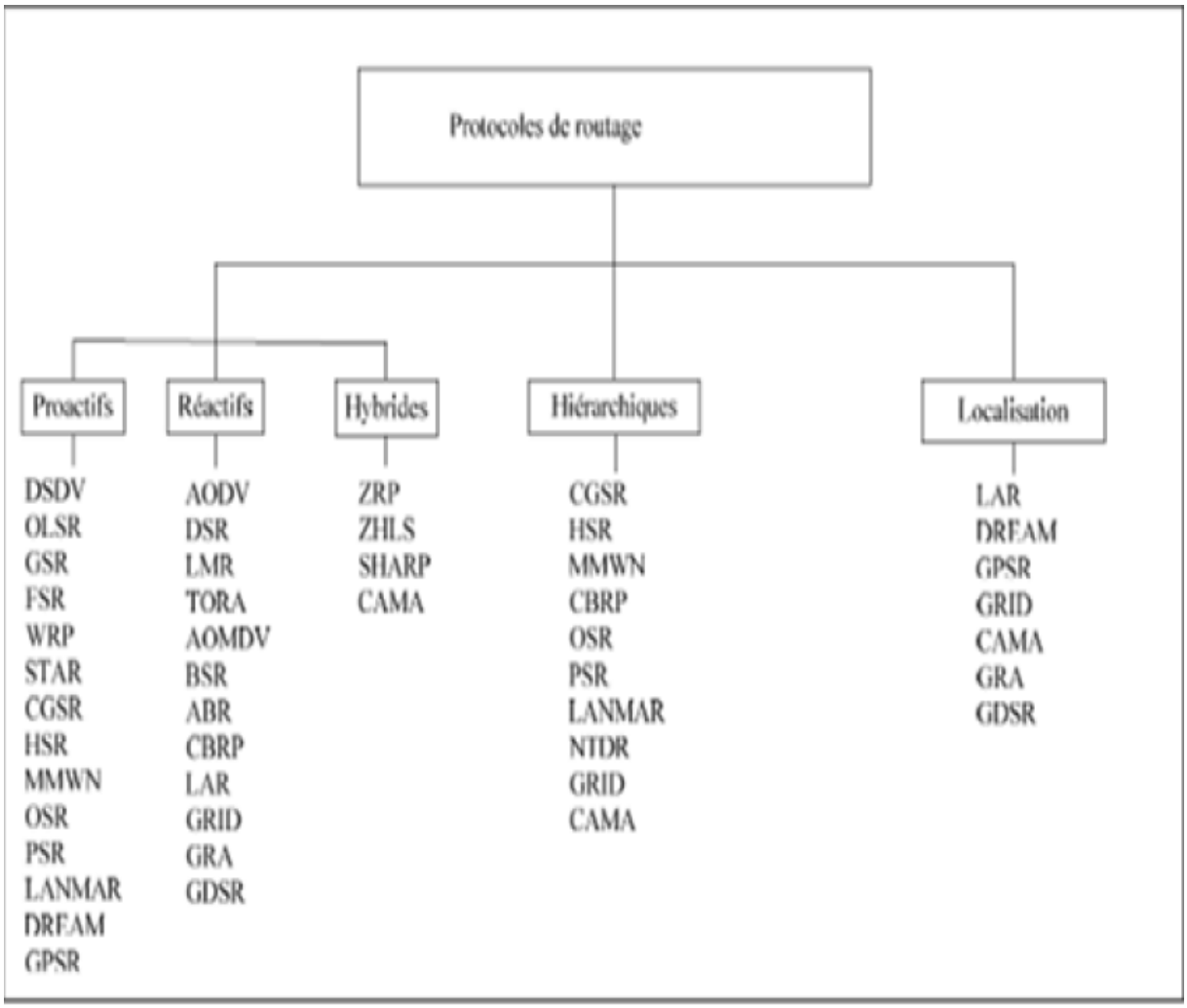


Figure 8 : Différentes classes des protocoles de routage. [15]

1.5.3. Les avantages et les inconvénients des protocoles de routage

Chaque type des protocoles de routage contient des avantages et des inconvénients, ces derniers ont des caractéristiques (avantages) que nous permettent de choisir le protocole de routage le plus adapté par les réseaux AD HOC. (Voir le tableau suivant).

Protocoles	Avantages	Inconvénients
Proactifs	• Pas de temps de réaction. • Adaptés aux réseaux denses de tailles moyennes. • Adaptés aux réseaux à forte mobilité.	• Trafic de contrôle important. • Capacité d'énergie de réseau Limitée. • Consommation énergétique plus importante.
Réactifs	• Trafic de contrôle faible. • Adaptés aux grands réseaux. • Consommation énergétique réduite.	• Temps de réaction long. • Problème en cas de forte mobilité des nœuds.
Hybrides	• Adaptable à tous les réseaux • Consommation énergétique réduite. • Bénéficier des avantages des deux approches précédentes.	• Recherche des routes complexes. • Temps de réaction long. • Cumuler les inconvénients des deux approches précédentes.

Table 1 : Bilan des différentes catégories de protocoles de routage.

1.6. Conclusion

Le réseau Ad hoc manifeste beaucoup de simplicité et assez d'avantages par rapport aux autres réseaux (filaires et cellulaires) par sa facilité de déploiement et son coût réduit. Cependant, les caractéristiques des réseaux ad hoc soulèvent de nouvelles problématiques qui sont spécifiques à ce type de réseau. Afin de satisfaire les besoins de toutes ces applications, de nouvelles fonctionnalités doivent être réalisées, plus particulièrement au niveau du routage de données et de la sécurité du routage. En effet l'absence d'une infrastructure centralisée fait du routage dans les réseaux ad hoc un problème très compliqué.

Dans le but d'assurer la connectivité du réseau malgré l'absence d'infrastructure et la mobilité des stations, chaque nœud est susceptible d'être mis à contribution pour participer au routage et pour retransmettre les paquets d'un nœud qui n'est pas en mesure d'atteindre sa destination, tout nœud joue ainsi le rôle de station et de routeur.

Donc, chaque nœud participe à une stratégie de routage qui lui permet de découvrir les chemins existants afin d'atteindre les autres nœuds du réseau. Finalement, nous avons présenté une classification des protocoles de routage dans les environnements mobiles, avec quelques exemples pour les protocoles de routage proactif, réactif et hybride qui ont été conçus pour les réseaux Ad hoc. Chacune de ces techniques est adaptée à un type particulier de réseau caractérisé par des caractéristiques spécifiques. De manière générale, les protocoles réactifs et proactifs présentent des performances différentes selon les caractéristiques du réseau. Dans le cas d'un réseau dense ou lorsque différentes paires de noeuds échangent fréquemment des données, un protocole réactif s'avère plus coûteux qu'un protocole proactif puisque la diffusion excessive de demandes de recherche de chemin concourt à une inondation/saturation du réseau. En revanche, un protocole réactif affiche de meilleures performances qu'un protocole proactif dès lors que le trafic généré par les noeuds est faible, puisqu'il ne surcharge pas inutilement le réseau par des vérifications continuelles de la localisation des noeuds. La sécurité des opérations de routage est de première importance pour le bon fonctionnement du réseau.

Et pour cela, Le prochain chapitre portera sur la notion de sécurité et les mécanismes utilisés pour sécuriser les protocoles de routage dans les réseaux ad hoc.

Chapitre 02

La Sécurité dans les réseaux ADHOC

2.1. Introduction

Les réseaux AD HOC sont de plus en plus populaires, ils vont être intégrer dans un futur proche toutes les situations de notre vie quotidienne. Une nécessité accrue s'est faite sentir pour rendre ces réseaux fiables et hautement sécurisés afin de protéger la vie privée des utilisateurs et offrir une bonne qualité de service pour les applications. Une tâche qui s'avère difficile et compliquée. Ce qui rend la tâche encore plus difficile est que les nœuds du réseau se chargent eux-mêmes de la fonction de routage des données. Favorisé par la nature vulnérable des communications sans fil, n'importe qui peut se connecter sur le réseau et écouter les messages de contrôle échangés. Il pourra ensuite les supprimer, les modifier, ou mener d'autres attaques plus complexes, ce qui met en danger tout le réseau. Les protocoles de routage proposés dans le cadre du travail du groupe MANET offre un acheminement optimal des données mais n'offre aucun système de sécurité. Dans ce cas-là, l'utilisation des systèmes de sécurité robustes et efficaces comme le cryptage par clé ou l'authentification sophistiquée qui consomment beaucoup de ressources ne donne pas toujours de bons résultats en pratique et peut affecter considérablement les performances du réseau.

Dans ce chapitre, nous allons mettre le point sur le problème de sécurité des protocoles de routage. Dans la première partie de ce chapitre nous introduisons les concepts et les terminologies fondamentales de la sécurité. La deuxième partie sera consacrée à l'étude des exigences de la sécurité, les différentes vulnérabilités liées aux protocoles de routage ad hoc, ainsi que les types d'attaques qui peuvent les menacées. Enfin, nous présenterons quelques approches utilisées pour la sécurité du routage ad hoc.

2.2. Définition de la sécurité

La sécurité d'information (SI) ou plus simplement sécurité informatique, est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires à la mise en place de moyens visant à empêcher l'utilisation non autorisée, le mauvais usage, la modification ou le détournement du système d'information. Assurer la sécurité du système d'information est une activité du management du système d'information.

2.3. Objectifs de la sécurité

La sécurité d'information représente un patrimoine essentiel de l'organisation. La sécurité informatique consiste à garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

La sécurité d'information vise les objectifs suivants (C.A.I.D.) :

- **Confidentialité**: seules les personnes autorisées peuvent avoir accès aux informations qui leur sont destinées (notions de droits ou permissions). Tout accès indésirable doit être empêché.
- **Authentification**: les utilisateurs doivent prouver leur identité par l'usage de code d'accès. Il ne faut pas mélanger identification et authentification : dans le premier cas, l'utilisateur n'est reconnu que par son identifiant, tandis que dans le deuxième cas, il doit fournir un mot de passe ou un élément que lui seul connait. Cela permet de gérer les droits d'accès aux ressources concernées et maintenir la confiance dans les relations d'échange.
- **Intégrité** : les données doivent être celles que l'on attend, et ne doivent pas être altérées de façon fortuite, illicite ou malveillante. En clair, les éléments considérés doivent être exacts et complets. Cet objectif utilise généralement des méthodes de calculs de checksum ou de hachage.

- **Disponibilité** : l'accès aux ressources du système d'information doit être permanent et sans faille durant les plages d'utilisation prévues. Les services et ressources sont accessibles rapidement et régulièrement. **[2]**

D'autres aspects peuvent aussi être considérés comme des objectifs de la sécurité des systèmes d'information, tels que :

- **La traçabilité** (ou « **preuve** ») : garantie que les accès et tentatives d'accès aux éléments considérés sont tracés et que ces traces sont conservées et exploitables.
- **La non-répudiation** et **l'imputation** : aucun utilisateur ne doit pouvoir contester les opérations qu'il a réalisées dans le cadre de ses actions autorisées et aucun tiers ne doit pouvoir s'attribuer les actions d'un autre utilisateur.

Une fois les objectifs de la sécurisation déterminés, les risques pesant sur chacun de ces éléments peuvent être estimés en fonction des menaces. Le niveau global de sécurité des systèmes d'information est défini par le niveau de sécurité du maillon le plus faible. Les précautions et contre-mesures doivent être envisagées en fonction des vulnérabilités propres au contexte auquel le système d'information est censé apporter service et appui.

2.4. Les menaces liées aux réseaux

2.4.1. Les menaces liées aux réseaux sans fil

On peut classifier les attaques dans un réseau sans fil Wi-Fi en deux groupes principaux : les attaques passives et les attaques actives, qui sont bien évidemment plus dangereuses.

- **Les Attaques passives :** Le Sniffing et l'analyse du trafic permettent à un intrus de prendre connaissance de la transmission des messages sur le réseau, et même de les exploiter ultérieurement d'une façon malhonnête. Dans un réseau sans fil, l'écoute passive est d'autant plus facile que le média air est difficilement maîtrisable. Bien souvent, la zone de couverture radio d'un point d'accès déborde du domaine privé d'une entreprise ou d'un particulier. L'attaque passive la plus répandue est la recherche de point d'accès. Cette attaque (appelée Wardriving) est devenue le " jeu " favori de nombreux pirates informatiques. Les points d'accès sont facilement détectables grâce à un scanner (portable équipé d'une carte WI-FI et d'un logiciel spécifique de recherche de points d'accès). Ces cartes Wi-Fi sont équipées d'antennes directives permettant d'écouter le trafic radio à distance dans la zone de couverture du point d'accès. Il existe deux types de scanners : les passifs (Kismet, Wi-Fiscanner, Prismstumbler...) ne laissant pas de traces (signatures), quasiment indétectables et ceux actifs (Netstumbler, Dstumbler) détectables en cas d'écoute, parce qu'ils envoient des " probe request ". Seul Netstumbler fonctionne sous Windows, les autres fonctionnent sous Linux. Les sites détectés sont ensuite indiqués par un marquage extérieur (à la craie) suivant un code (warchalking) : **[19]**

Deux demi-cercles opposés désignent ainsi un réseau ouvert offrant un accès à Internet	Un rond signale la présence d'un réseau sans fil ouvert sans accès à un réseau filaire	Un W encerclé met en évidence la présence d'un réseau sans fil correctement sécurisé
		

Figure 9 : Signification du Wardriving.

Une première analyse du trafic permet de trouver le SSID (nom du réseau), l'adresse MAC du point d'accès, le débit, le type de cryptage utilisé et la qualité du signal associés à un GPS, ces logiciels permettent de localiser (latitude, longitude) ces points d'accès. A un niveau supérieur, des logiciels (Aisnort ou Wepcrack) permettent en quelques heures (suivant le trafic), de déchiffrer les clés WEP et ainsi avec des outils d'analyse de réseaux conventionnels la recherche d'informations peut aller plus loin. Le pirate peut alors passer à une attaque dite active.

- **Les Attaques actives** : Les attaques les plus courantes sont définies comme suit : la modification, l'insertion, la modification et la suppression de messages, nécessitent une capture préalable du trafic durant la transmission vers les destinations. Une fois capturés, l'attaquant peut modifier ou insérer de fausses informations dans les messages avant de les rediffuser. Il peut aussi les supprimer. Les rejeux permettent à un attaquant de rejouer des sessions en remplaçant l'endroit où l'instant d'émission des messages. Les dénis de service (DoS). Une attaque DoS traditionnelle consiste à surcharger volontairement les connexions réseau en envoyant une quantité excessive de donnés jusqu'à la saturation de la bande passante. Ceci implique l'arrêt du traitement des donnés en entrée, ainsi le système sera paralysé. Le déni de service réseau est souvent l'alternative à d'autres formes d'attaques car dans beaucoup de cas il est plus simple à mettre en œuvre, et nécessite moins de connaissances et est moins facilement traçable qu'une attaque directe. Cette attaque a pour but d'empêcher des utilisateurs légitimes d'accéder à des services en les saturant avec de fausses requêtes. Elle se base généralement sur des " bugs " logiciels. Dans le domaine du Wi-Fi, cela consiste notamment à bloquer des points d'accès soit en l'inondant de requêtes de dés association ou de dés authentification (programme Airjack), ou plus simplement en brouillant les signaux hertziens. Plus généralement, les attaques DoS visent l'accès à un système, un réseau, une application ou une information pour un utilisateur légitime. Spoofing (usurpation d'identité). L'usurpation d'identité est réalisée par le Spoofing IP ou le Spoofing ARP etc. C'est une technique permettant à un pirate d'envoyer à une machine des paquets semblant provenir d'une adresse IP autre que celle de la machine du pirate. Cette attaque permet à un terminal d'être identifié, ou authentifié, à travers d'autres terminaux,

Comme une source légitime. C'est le cas où un attaquant communique avec une fausse identité. L'IP spoofing n'est pas pour autant un changement d'adresse IP. Plus exactement il S'agit d'une mascarade (il s'agit du terme technique) de l'adresse IP au niveau des paquets émis, c'est-à-dire que les paquets envoyés sont modifiés afin qu'ils semblent provenir d'une autre machine. *Man in the middle (home au milieu).* Cette attaque consiste pour un réseau Wi-Fi, à disposer d'un point d'accès (PA) étranger à proximité d'autres points d'accès légitimes. Les stations désirant se connecter au réseau livreront au PA " félon " leurs informations nécessaires à la connexion. Ces informations pourront être utilisées par une station pirate. Il suffit tout simplement à une station pirate écoutant le trafic, de récupérer l'adresse MAC d'une station légitime et de son point d'accès, et de s'intercaler au milieu. **[19]**

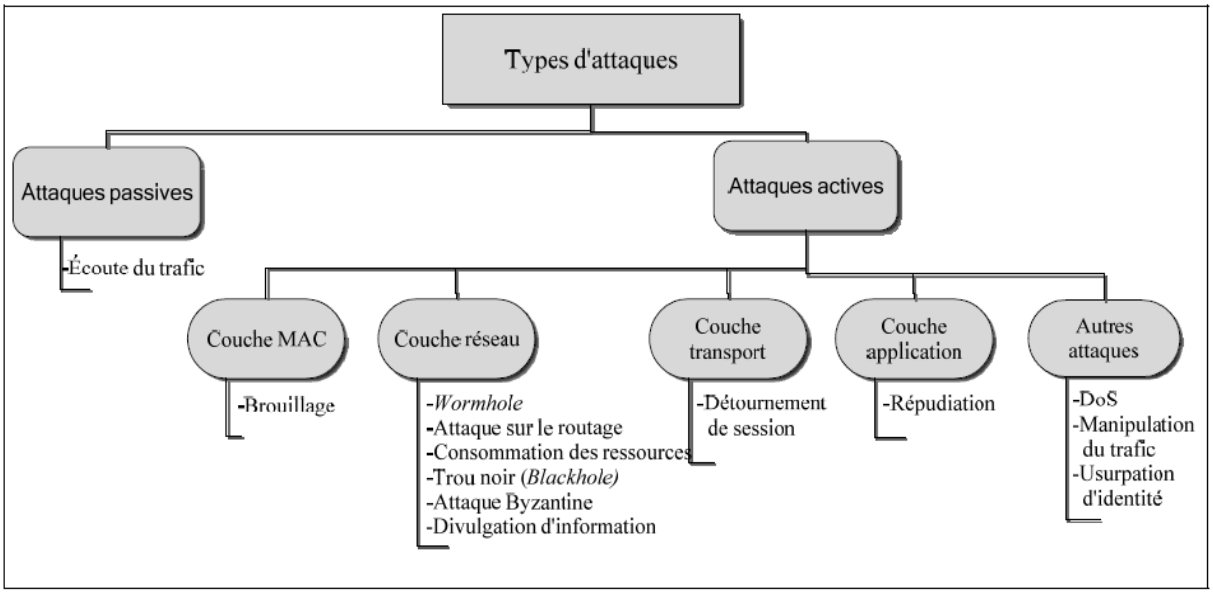


Figure 10 : Classifications des attaques. [2]

2.4.2. Les menaces liées aux réseaux AD HOC

Les attaques spécifiques aux réseaux ad hoc sont des attaques qualifiées de byzantines, et sont principalement l'attaque selfishness (Égoïsme) et l'attaque wormhole (Trou de ver). L'attaque Selfishness apparaît quand un nœud refuse de coopérer dans le processus de routage. En effet, les nœuds égoïstes refusent de relayer les messages de contrôle des autres nœuds et qui sont destinés à être diffusés dans le réseau entier. De telles attaques sont aussi appelées trou noir (Blackhole). Le trou noir est essentiellement employé quand l'intrus bloque systématiquement tous les messages reçus. Ce type d'attaques peut causer la rupture de connectivité entre plusieurs nœuds. Elle s'est déclinée en plusieurs variantes plus ou moins proches ayant des objectifs différents. Ainsi les boucles de routage (anglais : routing loops) permettent à un nœud de créer des boucles dans le réseau. De son côté, le trou gris (anglais : gray hole) ne laisse passer que les paquets de routage et détourne les données. Quant à black mail, il permet à un nœud malveillant d'isoler un autre nœud, etc. Il y a usurpation d'identité, comme nous le montre la Figure 13.

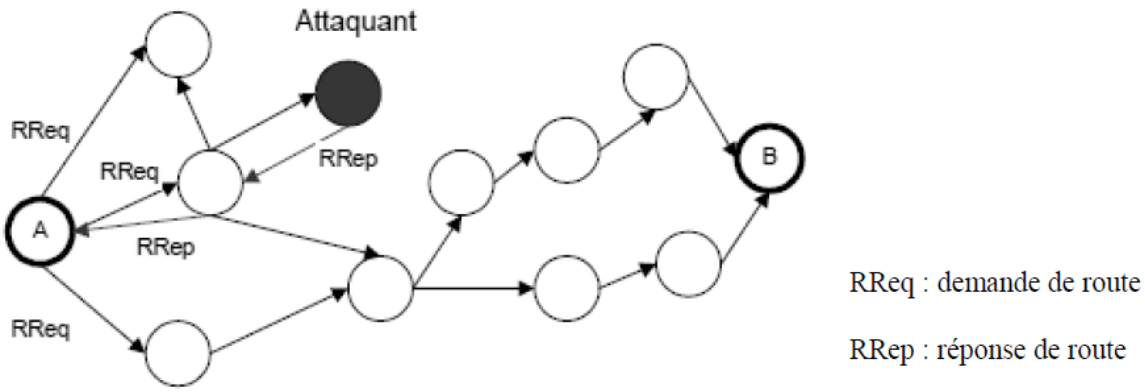


Figure 11 : Exemple de l'attaque du trou noir. [19]

Les attaques Wormhole (Trou de ver) impliquent la participation d'au moins deux nœuds, qui participent à créer un tunnel entre eux dans le but de réaliser un raccourci (wormhole) dans le réseau. Une fois le tunnel créé, les deux attaquants encapsulent les messages reçus et les échangent à travers le tunnel, en privant donc les nœuds intermédiaires de recevoir les messages de contrôle du routage. Le premier nœud retransmet des paquets via le tunnel à l'autre bout pour les réinsérer corrompus dans le réseau. Cette attaque peut alors faciliter la mise en place d'une autre attaque nommée rushing attack. Cette dernière consiste à profiter du fait que dans la plupart des protocoles de routage, lors de la découverte de routes, c'est la première requête qui arrive aux nœuds intermédiaires qui est transmise. L'objectif pour l'attaquant est alors de faire passer ses requêtes avant les autres.

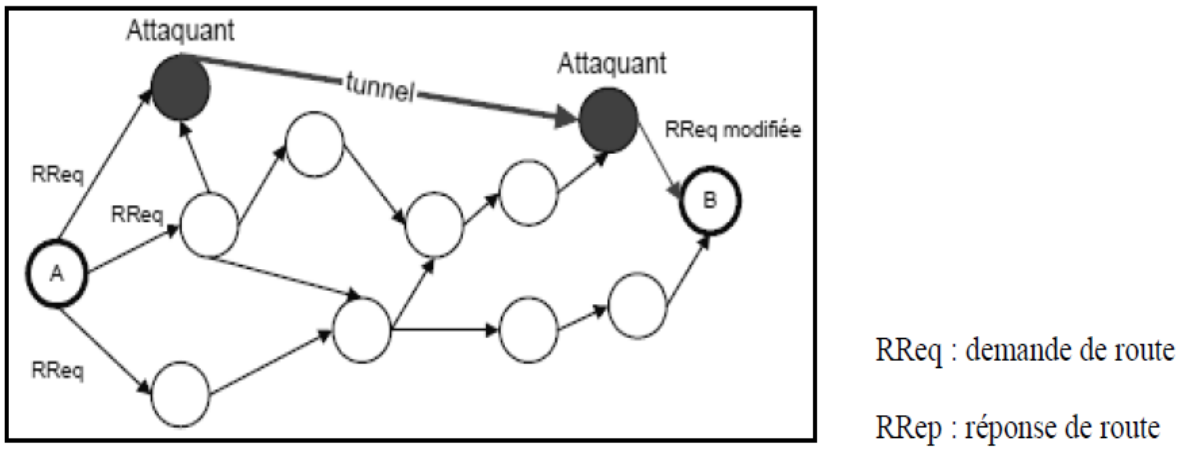


Figure 12: Exemple de l'attaque du trou de ver. [19]

Ces attaques représentent un défi sérieux qui est capable de causer de sérieux problèmes dans les protocoles de routage ad hoc. Tous ces types d'attaques ont comme conséquences majeures, la modification ou la suppression des messages de routage, avec comme conséquence, des routes erronées. Le processus de routage se trouve donc ralenti, ou devient complètement perturbé.

Attaques	Définition	Solutions proposées
Wormhole	Un attaquant pourrait rediriger le trafic entre deux zones géographiquement éloignées pour créer un vertex dans la topologie et ainsi avoir une bonne position géographique pour Contrôler le trafic qui passe par lui.	*Packet Leashes (Hu, Perrig et Johnson, 2003).*
Attaque de routage	Un nœud malicieux pourrait perturber le fonctionnement d'un protocole de routage en modifiant les informations de routage, fabriquer les fausses informations de routage ou usurper L'identité d'un autre nœud.	SEAD (Perkins et Bhagwat, 1994), ARAN (Sanzgiri et al., 2002), ARIADNE (Hu, Perrig et Johnson, 2002), SAODV (Zapata, 2002).
Brouillage *(Jamming)*	C'est une attaque classique sur la disponibilité du canal de communication grâce à la génération massive d'une grande Quantité d'interférence radio.	FHSS, DSSS (Wang et al., 2006).
Attaque trou noir (*Backhole attack*)	Le but de cette attaque est la falsification des informations de Routage ou le détournement du trafic.	(Ramaswamy et al., 2003).
Attaque sur les ressources	Les réseaux MANET sont caractérisés par des ressources limitées (batterie et bande passante). Une attaque sur les Ressources pourrait avoir des conséquences sur la disponibilité.	SEAD (Perkins et Bhagwat, 1994).
Attaque Byzantine	Grâce à cette attaque, un nœud malicieux altère les messages et pourrait créer des problèmes de boucle de routage, routage de paquets vers des chemins non optimaux, sélectionner les paquets à rejeter… Ce type d'attaque est difficile à détecter car Le réseau semble fonctionner correctement.	OSRP (Awerbuch et al., 2002), (Awerbuch et al., 2004).
DoS	Ce type d'attaque consiste à envoyer délibérément des Messages pour causer une saturation de la bande passante et paralyser le réseau.	SEAD (Perkins et Bhagwat, 1994), ARIADNE (Hu, Perrig et Johnson, 2002), SAODV (Zapata, 2002).
Répudiation	Ce type d'attaque a une conséquence sur l'intégrité des Communications entre les nœuds dans le réseau.	ARAN (Sanzgiri et al., 2002).
Usurpation d'identité	L'usurpation d'identité a pour but la falsification des informations relatives aux identités. Ce qui pourrait conduire à l'isolement de nœuds, l'échange de fausses informations de Routage et l'atteinte à la confidentialité et l'intégrité.	ARAN (Sanzgiri et al., 2002), SAODV (Zapata, 2002).

Table 2 : Les différentes attaques avec leurs solutions. [18]

SEAD: Secure Efficient Ad hoc Distance vector routing protocol.
SAODV: Secure Ad-Hoc On-demand Distance Vector routing.
ARAN: Authenticated Routing for Ad-Hoc Networks.
ARIADNE: A Secure On-Demand Routing Protocol for Ad-Hoc Networks.
FHSS: Frequency-Hopping Spread Spectrum.
OSRP: On-demand Secure Routing Protocol.
SMT: Secure Message Transmission Protocol.
SRP: Secure Routing Protocol for Mobile Ad-Hoc Network.
DSSS: Direct-Sequence Spread Spectrum.

2.5. Solution de sécurité

Il n'est pas dans nos objectifs de citer les différents types d'attaques sur les environnements sans fil comme nous venons de le voir, mais nous envisageons d'en proposer des solutions qui permettent soit d'empêcher définitivement l'attaque, soit d'en amoindrir l'effet. Dans cette vision, Donc nous présentons les différentes solutions possibles :

2.5.1. La cryptographie

Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre intelligibles sans une action spécifique. Le verbe crypter est parfois utilisé mais on lui préfèrera le verbe chiffré. La cryptologie est essentiellement basée sur l'arithmétique : Il s'agit dans le cas d'un texte de transformer les lettres qui composent le message en une succession de chiffres (sous forme de bits dans le cas de l'informatique car le fonctionnement des ordinateurs est basé sur le binaire), puis ensuite de faire des calculs sur ces chiffres pour : **[21]**

- D’une part les modifier de telle façon à les rendre incompréhensibles. Le résultat de cette modification (le message chiffré) est appelé cryptogramme (en anglais ciphertext) par opposition au message initial, appelé message en clair (en anglais plaintext).
- Faire en sorte que le destinataire saura les déchiffrer.

Le fait de coder un message de telle façon à le rendre secret s'appelle chiffrement. La méthode inverse, consistant à retrouver le message original, est appelée déchiffrement.

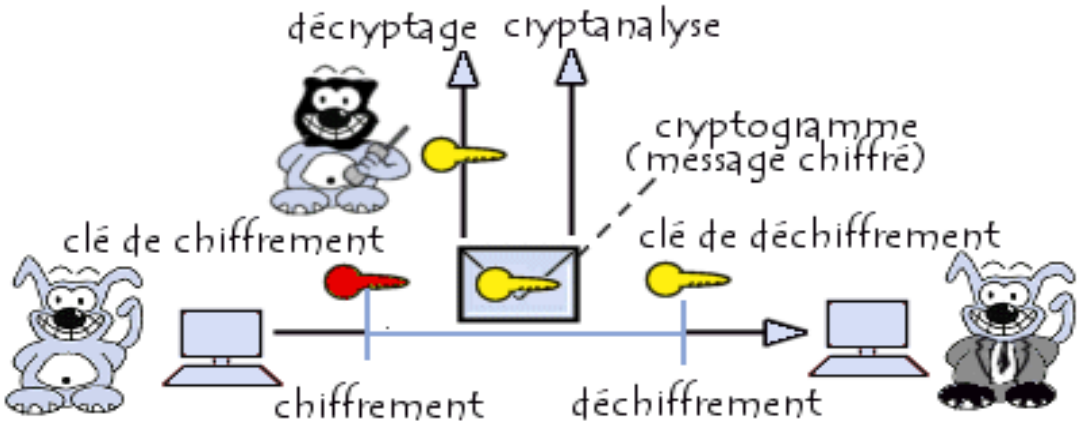


Figure 13 : Schéma représentatif sur la cryptographie. [21]

Le chiffrement se fait généralement à l'aide d'une clé de chiffrement, le déchiffrement nécessite quant à lui une clé de déchiffrement. On distingue généralement deux types de clés :

- **Les clés symétriques :** il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
- **Les clés asymétriques :** il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

2.5.2. Les fonctions de hachage

Une fonction de hachage est typiquement une fonction qui à une très grande taille (théoriquement infini) et de nature très diversifiée, elle va renvoyer des résultats aux spécifications précises (en général des chaînes de caractère de taille limitée ou fixe) optimisées pour des applications particulières. Les chaînes permettent d'établir des relations (égalité, égalité probable, non-égalité, ordre...) entre les objets de départ sans accéder directement à ces derniers, en général soit pour des questions d'optimisation (la taille des objets de départ nuit aux performances), soit pour des questions de confidentialité. **[22]**

En terme très concret, on peut voir une fonction de hachage (non cryptographique) comme un moyen de replier l'espace de données que l'on suppose potentiellement très grand et très peu rempli pour le faire entrer dans la mémoire de l'ordinateur. En revanche, une fonction de hachage cryptographique est ce que l'on appelle une fonction à sens unique, ce qui veut dire que le calcul de la fonction de hachage est facile et rapide tandis que le calcul de sa fonction inverse est infaisable par calcul et donc non calculable en pratique. Grâce à la valeur de hachage (le hash), on peut discriminer deux objets apparemment proches, ce qui peut être utilisé pour garantir l'intégrité des objets, autrement dit leur non modification par une erreur ou un acteur malveillant.

2.5.3. Les chaînes de hachage

Les chaînes de hachage sont basées sur les fonctions de hachage à sens unique. Une chaîne de hachage de longueur N est construite en appliquant une fonction de hachage N fois sur une valeur aléatoire appelé XN. La valeur XN est appelée valeur racine de la chaîne de hachage. On définit une chaîne de hachage en utilisant la fonction de hachage h par :

$$\begin{cases} h_i(Y) = h(h_{i-1}(Y)) \\ h_O(Y) = X_N \end{cases}$$

Où hi(Y) est le résultat de l'utilisation répétée i fois de la fonction de hachage à la valeur initiale Y. La valeur finale de hachage de la chaîne de hachage XO = hN (XN) est obtenue en appliquant la fonction de hachage N fois.

Le récepteur applique une seule fois la fonction de hachage pour vérifier la valeur de hachage reçue. Puisque la fonction de hachage est à sens unique, seulement l'utilisateur qui a créé la chaîne de hachage peut générer la valeur de hachage qui précède la valeur envoyée. **[20]**

2.5.4. La signature numérique

La signature numérique est définie comme des « données ajoutées à un message », ou transformation cryptographique d'un message, permettant à un destinataire de :

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

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

Étapes de signature d'un message :

La signature numérique comprend deux étapes :

- Évaluation du condensé de message : l'émetteur commence par générer un condensé, qui est une représentation réduite et unique du message complet, à l'aide d'une fonction de hachage.
- Signature du condensé : l'émetteur chiffre ce condensé avec un algorithme asymétrique à l'aide de sa clé privée. Il obtient une signature électronique qu'il appose au message original avant d'émettre l'ensemble, message et signature, sur le réseau. **[23]**

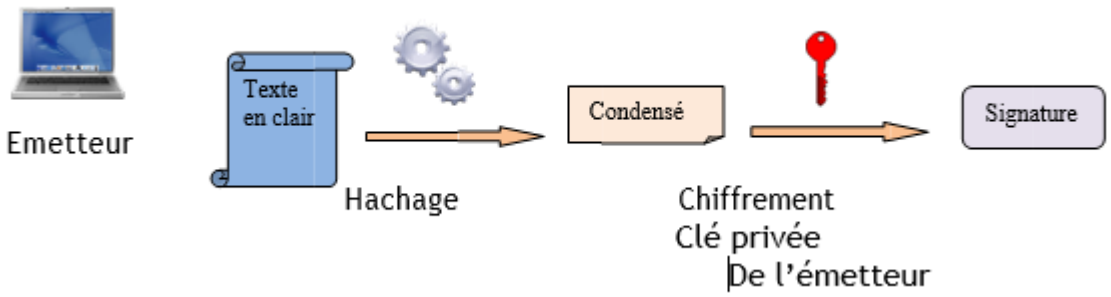


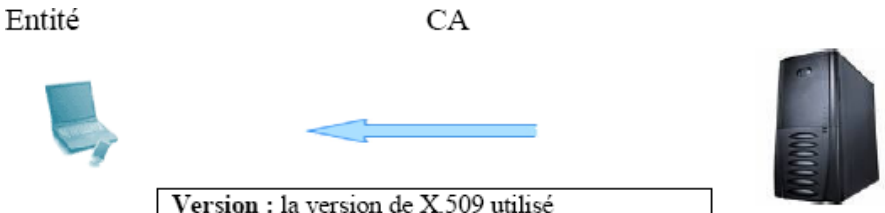
Figure 14 : Signature d'un message. [23]

2.5.5. Certificats électroniques

Un certificat est un élément d'information qui prouve l'identité du propriétaire d'une clé publique. Les certificats sont signés et transmis de façon sécuritaire par un tiers de confiance appelé autorité de certification (Certificate Authority, ou CA). L'autorité de certification est chargée de délivrer les certificats, de leur assigner une date de validité, ainsi que de révoquer éventuellement des certificats avant cette date en cas de compromission de la clé (ou du propriétaire). **[2]**

La structure des certificats est normalisée par le standard X.509 de l'UIT, qui définit les informations contenues dans le certificat :

- Version.
- Numéro de série de l'autorité de certification.
- Algorithme de signature du certificat.
- Le nom de l'autorité de certification.
- Le nom du propriétaire du certificat.
- La date de validité du certificat.
- Le propriétaire du certificat.
- La clé publique du propriétaire.



Version : la version de X.509 utilisé
Serial Number : numéro de séquence du certificat (propre à chaque autorité de certification)
Signature Algorithm : algorithme utilisé pour signer
Issuer : le nom du CA qui a émit le certificat
Validity : période de validité
Subject : le nom du propriétaire de ce certificat
Public Key : clé publique de l'utilisateur
Subject Public Key Info : des informations sur la clé publique (type, longueur…etc.)
Signature numérique : signature du CA sur l'ensemble des champs précédents.

Figure 17 : Les informations contenues dans le certificat. [2]

2.5.6. La réputation

Chaque entité réseau encourage la collaboration d'autres entités en utilisant une métrique de coopération appelée réputation. La métrique de réputation est calculée sur la base des données recueillies localement par chaque nœud et peut se baser optionnellement sur l'information fournie par d'autres nœuds du réseau impliqués dans des échanges de messages avec les nœuds surveillés. Une note est attribuée à chaque entité, cette note sera augmentée chaque fois que l'entité participe au routage. Basé sur la réputation, un mécanisme de punition est adopté comme système de discussion pour empêcher un comportement égoïste en refusant graduellement les services de communication aux entités qui se conduisent mal. Les nœuds prouvés comme malveillants sont exclu du réseau. Cette décision est prise par une autorité centrale.

2.6. Conclusion

La sécurité du routage dans les réseaux ad hoc est complexe. Nous pouvons constater qu'il n'existe pas de solution complète pour remédier à ce problème. Plusieurs schémas de sécurité ont été proposés chacun de ces schémas à ses propres besoins et contraintes qui s'imposent pour atteindre la sécurité voulue. Les protocoles se basant sur un mécanisme cryptographique requièrent un schéma de distribution et de gestion de clés. Les protocoles se basant sur la réputation incluent une nouvelle métrique (degré de fiabilité du chemin) pour sélectionner une route vers la destination. Enfin il n'existe pas un schéma résistant à toutes les attaques et les vulnérabilités. En ce qui concerne les problèmes de sécurité, il faudra probablement toujours trouver des systèmes de plus en plus complexes pour faire face à la ténacité et l'ingéniosité des pirates qui chercher ont toujours à relever le défi. Donc, La solution optimale de sécurité n'existe pas. D'une part, les techniques proposées offrent des solutions partielles et adaptées à quelques failles seulement. Les techniques les plus élaborées sont très coûteuses. Les réseaux ad hoc constituent par leur nature, un formidable challenge pour la sécurité informatique.

Et pour cela, Dans le prochain chapitre on va faire une étude approfondis sur le protocole de routage OLSR et les mécanismes utilisés pour le sécuriser. OLSR est une optimisation d'un protocole d'état de liaison pour les réseaux mobiles ad hoc. Premièrement, il réduit la taille du paquet de contrôle ; au lieu de tous les liens, il déclare qu'une partie des liens avec ses voisins. Deuxièmement, il minimise les inondations de la circulation par ce contrôle en utilisant uniquement les nœuds sélectionnés, pour diffuser son message dans le réseau. Seuls ces nœuds qui peuvent retransmettre ses messages diffusés. Cette technique réduit considérablement le nombre de retransmissions dans une procédure d'inondation ou de diffusion. Le protocole est conçu pour fonctionner de manière complètement distribuée et n'a donc pas à dépendre de toute entité centrale. Le protocole ne nécessite pas une transmission fiable pour ses messages de contrôle et Son innovation réside dans sa façon d'économiser les ressources radio lors des diffusions.

Chapitre 03

Etude de La Sécurité du protocole du réseau OLSR

3.1. Introduction

OLSR (Optimized Link State Routing Protocol), est un protocole de routage destiné aux réseaux mobiles. Comme étant un protocole proactif, OLSR maintient en temps réelles ses tables de routage. Ce qui permet à chaque station de travail de connaître les nœuds avec lesquels elle peut communiquer ainsi que les routes optimales qui mènent vers ces nœuds. Ceci est réalisé grâce à l'utilisation des relais multipoints. Ce protocole est l'objet de notre travail. Nous l'utiliserons pour construire notre schéma sécurisé.

Dans ce chapitre, nous allons présenter le protocole de routage OLSR, en donnant une description détaillée de ce protocole et son principe de fonctionnement. Nous décrivons par la suite, ses vulnérabilités et ses failles et nous finirons par une étude synthétique des solutions proposées pour sécuriser OLSR. Ainsi que les types d'attaques qui peuvent les menacées.

3.2. Définition du protocole OLSR

Le protocole **« Optimized Link State Routing »** développé par Thomas Clausen et Philippe Jacquet dans le cadre du projet HIPERCOM, Son fonctionnement est basé sur l'algorithme à état de liens. Chaque nœud doit déterminer l'ensemble de ses voisins. Pour cela périodiquement, ils transmettent des paquets, dits Hello, pour se faire connaître. Ce type de paquet comprend la totalité de la base de liens connue par l'émetteur du paquet. La base de liens d'un nœud regroupe l'ensemble des nœuds lui ayant transmis un paquet Hello. A la réception des paquets Hello, chaque nœud dans le réseau connaît les nœuds situés dans son voisinage immédiat mais également à deux sauts. Une fois les voisins découverts, les nœuds peuvent échanger les informations sur leur voisinage pour former la topologie du réseau. Cette fonction est attribuée à des nœuds particuliers sélectionnés parmi ses voisins à un saut. Ces nœuds sont appelés relais multipoints **« MPRs »** et sont les seuls capables de transmettre les informations de routage. Chaque nœud sélectionne un ensemble de **MPRs** relayant les informations de routage à l'ensemble des nœuds situés à deux sauts. Chaque **MPR** transmet périodiquement la liste des nœuds qui l'ont choisi comme **MPR**. Un tel paquet est seulement relayé par les nœuds sélectionnés en tant que **MPRs**. Une fois la topologie connue par l'ensemble des nœuds du réseau, il suffit d'appliquer un algorithme, de type Dijkstra, pour déterminer les routes vers l'ensemble des nœuds distants. Chaque nœud connaît, ainsi, les routes les plus courtes vers les autres nœuds du réseau **[25].**

Protocoles	Avantages	Inconvénients
OLSR	-Il offre des fonctionnalités très intéressantes tout en recherchant des routes optimales en termes de nombre de sauts.	-Le problème actuel d'OLSR est celui de la sécurité

Table 3 : Avantages et inconvénients du protocole OLSR

3.3. Fonctionnement de l'OLSR

Le protocole OLSR est une variation du LSR « Link State Routing » spécialement conçu pour les MANET. Contrairement au LSR où tous les nœuds sont indifférenciés, l'optimisation d'OLSR est d'utiliser des relais multipoints (MPR). Les MPR sont des nœuds choisis qui expédient des messages de diffusion pendant le processus d'inondation. Ils sont les seuls à déclarer leurs liens et sont sélectionnés par les autres nœuds de manière que ceux-ci puissent atteindre n'importe qui en deux sauts. Cette technique réduit sensiblement la surcharge due aux messages par rapport à un mécanisme classique d'inondation, où chaque nœud retransmet chaque message quand il reçoit la première copie du message. Dans OLSR, l'information d'état de lien est produite seulement par des nœuds élus comme MPR, ainsi, une deuxième optimisation est réalisée en réduisant au minimum le nombre des messages de contrôle inondés dans le réseau. Comme troisième optimisation, un nœud de MPR doit rapporter seulement des liens entre lui-même et ses sélecteurs.

Les deux principales fonctionnalités d'OLSR sont :

- La découverte des voisins.
- La diffusion de la topologie.

Le protocole est conçu pour fonctionner de manière complètement distribuée et n'a donc pas à dépendre de toute entité centrale. Le protocole ne nécessite pas une transmission fiable pour ses messages de contrôle : chaque nœud envoie ses messages de contrôle périodiquement, messages qui peuvent subir une perte de certains des paquets, ce qui arrive très souvent dans les réseaux radio en raison de collisions ou d'autres problèmes de transmission.

3.3.1. Détection des voisins

Chaque nœud doit détecter les nœuds voisins avec lesquels il a un lien direct et bidirectionnel. Les incertitudes sur la propagation radio peuvent rendre certains liens unidirectionnels. Par conséquent, tous les liens doivent être contrôlés dans les deux directions, afin d'être considérés comme valides.

Pour accomplir cela, chaque nœud diffuse périodiquement ses messages HELLO contenant les informations sur ses voisins et leur état de lien. Ces messages de contrôle sont transmis dans le mode de diffusion. Ils sont reçus par tous les voisins situés à un saut, mais ils ne sont pas relayés à des nœuds supplémentaires.

Un des messages HELLO contient :

- La liste des adresses des voisins pour lesquels il existe un lien bidirectionnel valide.
- La liste des adresses des voisins qui sont entendues par ce nœud (un HELLO a été reçu), mais le lien n'est pas encore validé comme bidirectionnel : si un nœud trouve sa propre adresse dans un message HELLO, il considère le lien du nœud expéditeur comme bidirectionnel.

3.3.2. Relais multipoints ou MPR

L'idée des relais multipoints est de minimiser l'inondation de paquets de diffusion dans le réseau en réduisant les retransmissions en double vers un même nœud. Chaque nœud dans le réseau sélectionne un ensemble de nœuds dans son voisinage, qui retransmet ses paquets. Cet ensemble de nœuds voisins sélectionné est appelé le relais multipoint de ce nœud ou MPR.

Les voisins du nœud N qui ne sont pas dans son ensemble MPR, peuvent lire et traiter le paquet, mais ne peuvent pas retransmettre le paquet de diffusion reçu à partir du nœud N. Pour cela, chaque nœud maintient une liste de ses voisins qui sont appelés les sélecteurs de MPR de ce nœud. Chaque message de diffusion provenant de ces sélecteurs MPR d'un nœud est supposé être retransmis par ce nœud. Cet ensemble peut changer au fil du temps, ce qui est indiqué par le sélecteur de nœuds dans leurs messages.

Chaque nœud sélectionne ses relais multipoints parmi ses voisins situés à un saut. Un saut correspond à un nœud dont la distance est la plus proche du nœud principal. Les relais multipoints sont choisis de manière à couvrir tous les nœuds qui sont situés à deux nœuds de distance. L'ensemble de relais multipoints d'un nœud N, noté MPR (N), est un sous-ensemble arbitraire du nœud de N qui satisfait la condition suivante : chaque nœud dont la distance est à deux sauts de N doit avoir un lien bidirectionnel vers les relais multipoints du nœud N. La figure montre la sélection de relais multipoints autour du nœud N.

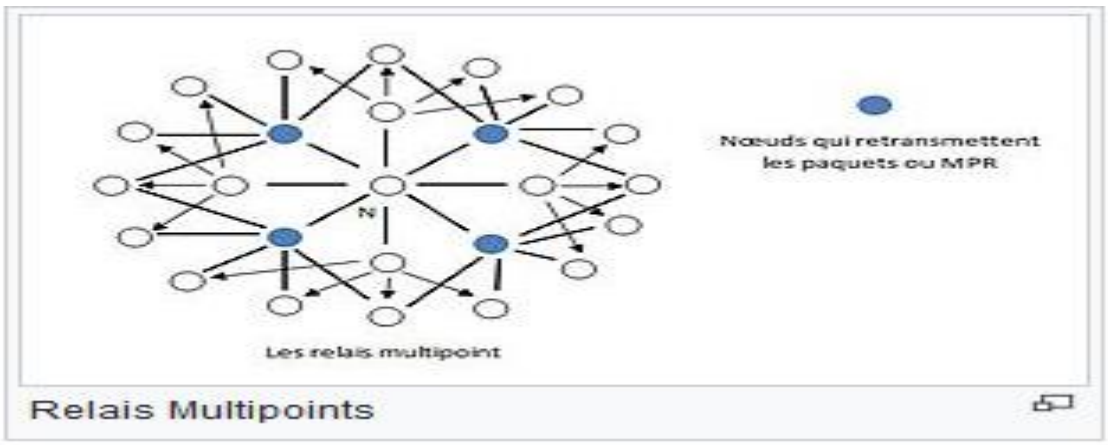


Figure 16 : Les relais multipoint. [2]

OLSR repose sur la sélection des relais multipoints, et calcule ses routes vers toutes les destinations connues à travers les nœuds. Les nœuds MPR sont choisis comme des nœuds intermédiaires dans le chemin. Pour mettre en œuvre ce schéma, chaque nœud dans le réseau envoie périodiquement des informations à ses voisins qui ont été choisis comme relais multipoint. Dès réception de l'information sur les sélecteurs MPR, chaque nœud calcule et met à jour ses itinéraires pour chaque destination connue. Par conséquent, la route est une séquence de sauts à travers les relais multipoints de la source à la destination.

Les relais multipoints sont choisis parmi les voisins à un saut avec un lien bidirectionnel. Ainsi, l'itinéraire en passant par les relais multipoints évite automatiquement les problèmes associés au transfert de données par paquets sur les liens unidirectionnels.

Pour le calcul d'itinéraire, chaque nœud calcule sa table de routage en utilisant un "algorithme de plus court chemin hop" basé sur la topologie du réseau partiel qu'il a appris.

3.3.2.1. Exemple de sélection des MPR

La sélection des MPR est le point clé dans le protocole OLSR. La sélection du MPR se fait en choisissant le nœud voisin à un saut. S'il y a plusieurs nœuds, le nœud choisi est alors celui qui a le plus de voisins. Le tableau montre comment le nœud B sélectionne un MPR, basé sur le réseau représenté dans la figure :

Noeud	Nœud à 1 saut	Nœud à 2 sauts	MPR
B	**A,C,F,G**	**D,E**	**C**

Table 4 : La sélection du MPR.

Si on prend le nœud B, les nœuds C et F couvrent l'ensemble des nœuds voisins à deux sauts. Cependant, le nœud C est sélectionné en tant que MPR car il a 5 voisins alors que le nœud F en possède 4 (on dit alors que le degré de C est supérieur au degré de F).

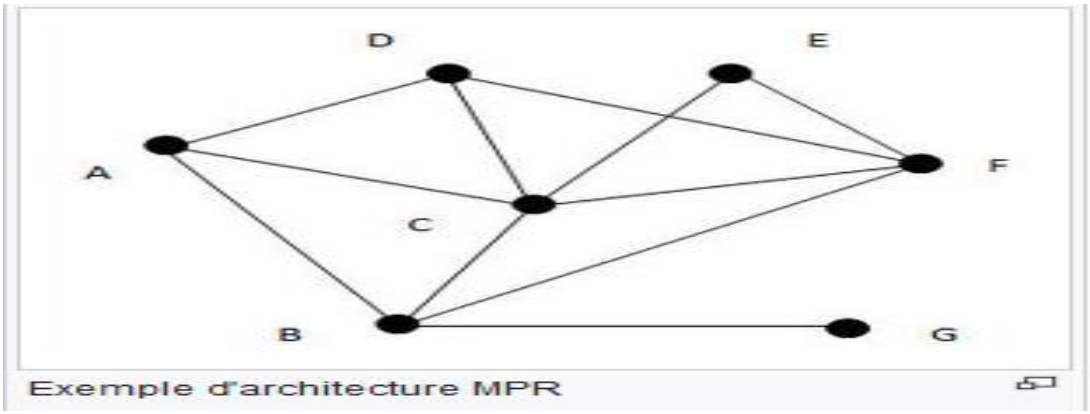


Figure 17 : Exemple de sélection d'un MPR. [2]

3.3.3. Format du paquet OLSR

Le protocole OLSR définit un format général du paquet, donné sur la figure ci-dessous. Ce format est unique pour tous les messages circulant sur le réseau. Chaque paquet peut contenir plusieurs messages identifiés par un type. Ceci permet d'envoyer plusieurs informations à un nœud en une seule transmission. Selon la taille de MTU (Maximum Transfer Unit), un nœud peut ajouter différents types de messages et les transmettre ensemble. Par conséquent différents types de messages peuvent être émis ensemble mais traités et retransmis différemment dans chaque nœud. Par exemple les messages de type HELLO ne sont pas relayés tandis que ceux de type TC le sont.

Quand un nœud reçoit un paquet, il examine les entêtes des messages et en détermine le type selon la valeur du champ message type. La structure de base d'un paquet OLSR est la suivante:

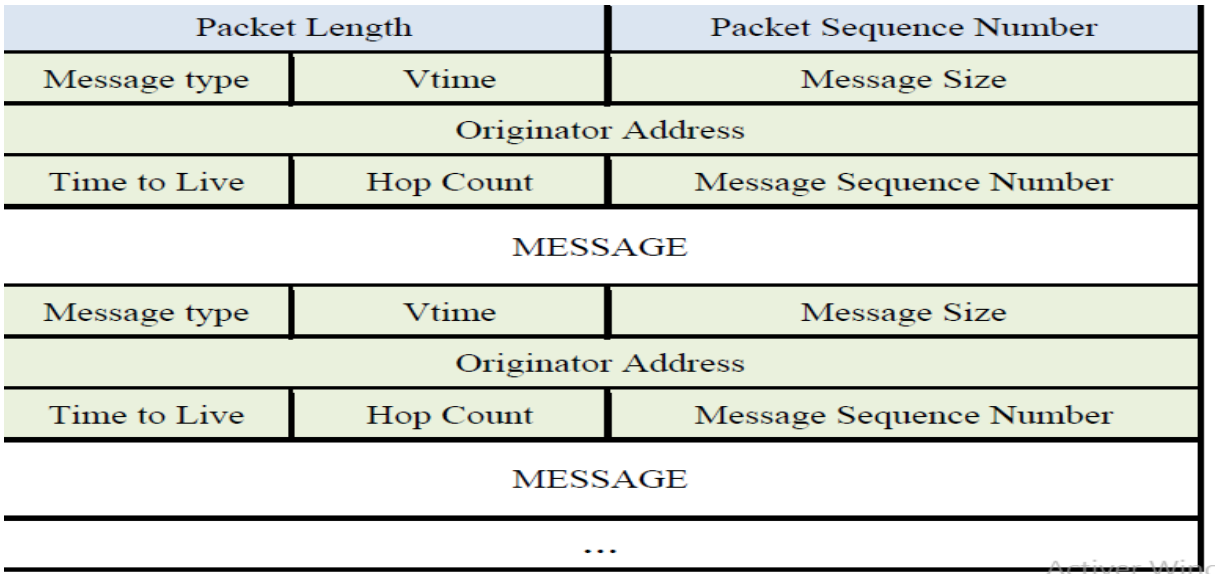


Figure 18 : Format d'un paquet OLSR.

Dans l'entête du paquet OLSR, on trouve :

• « **Packet Length** » **:** La longueur en octets du paquet.
• « **Packet Sequence Number (PSN)** » **:** Le numéro de séquence du paquet, il sera incrémenté à chaque paquet transmis.

Et dans l'entête du message OLSR, on trouve :

• « **Message type** » **:** Ce champ indique le type de message qui se trouve dans la partie "MESSAGE" (HELLO, TC, MID et HNA).
• « **Vtime** » **:** Durée de validité du message.
• « **Message Size** » **:** La taille du message exprimée en octets.
• « **Originator Address** » **:** L'adresse principale du nœud qui a produit le message ‘’MESSAGE, et elle ne doit être jamais changé dans la retransmission du paquet.
• « **Time to Live** » **:** Nombre maximum de sauts dont un message sera transmis et pour chaque passage par un nœud il décrémente par un 1.
• « **Hop Count** » **:** Nombre de sauts qu'un message a atteints. Au début, ceci est placé à 0 par l'initiateur du message et doit être incrémenté par 1 à chaque transmission.
• « **Message Sequence Number** » **:** Numéro d'identification unique pour chaque message. Ce nombre est incrémenté a 1 pour chaque transmission pour savoir si le message déjà reçu ou non.

3.3.4. Type de messages

3.3.4.1. Message Hello

Les messages HELLO sont utilisés pour la détection de voisin et calcul du MPR. Ils sont transmis périodiquement à tous les voisins à 1 saut. Les messages HELLO incluent le type de lien, la volonté du nœud de devenir MPR, des informations sur les voisins etc. Le type de lien dans ces messages indique si le lien est symétrique, asymétrique ou tout simplement perdu. Ces messages ne sont destinés qu'aux nœuds voisins (à un saut) de l'expéditeur afin de détecter les liens les interconnectant, ils ne doivent donc jamais être routés par un MPR. Il sert d'abord à découvrir l'ensemble du réseau. Il transmet ensuite l'état et le type de lien entre l'expéditeur et chaque nœud voisin.

Il contient également trois listes :

- Voisins qui ont été "entendus" mais pour lesquels une communication bidirectionnelle n'a pu être établie.
- Voisins avec qui le nœud a pu établir une liaison bidirectionnelle.
- Les nœuds désignés comme MPR par le nœud originaire du message HELLO.

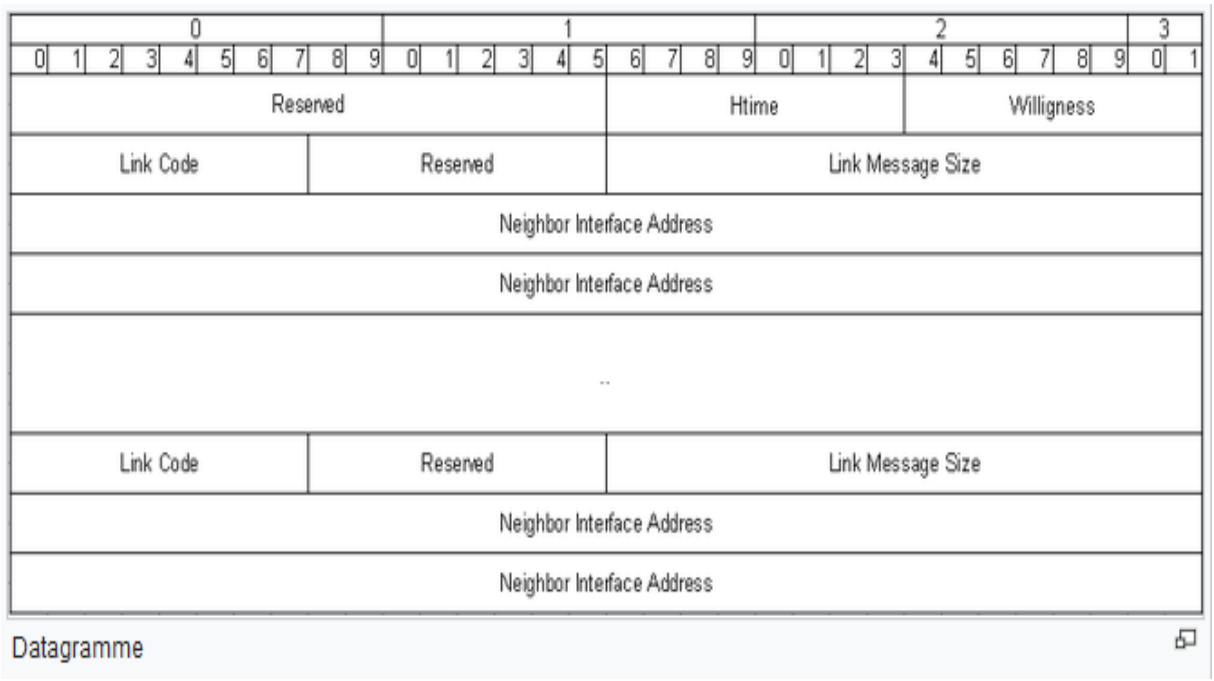


Figure 19 : Format d'un message HELLO. [26]

- « **Reserved** » : Ce champ doit contenir « 0000000000000000 ».
- « **Htime** » : Intervalle d'émission des messages HELLO avant la transmission du prochain message HELLO.
- « **Willigness** » : volonté du nœud entre 0 et 7, donc un nœud avec une bonne volonté est toujours un MPR.
- « **Link Code** » : Code identifiant le type de lien (pas d'information, symétrique, asymétrique, etc.) entre l'expéditeur et les interfaces listées.
- « **Link message size** » : la taille du message de lien en bits.
- « **Neighbor interface address** » : c'est l'adresse de l'interface du nœud voisin.

Donc, un message HELLO sert à la découverte locale de la topologie réseau avec :

- Sensation de lien.
- Détection du voisin.
- Signalisation de choix MPR.

3.3.4.2. Message TC (topology control)

Ces messages sont utilisés pour construire la table de routage. Ce sont des messages d'état de liaison, diffusés périodiquement dans des réseaux entiers.

- Expéditeur : seuls les MPR envoient des messages TC.
- Destinataire : adresse de broadcaste.
- Fonction : le message TC permet au MPR de transmettre la liste de ses voisins qui l'ont choisi comme MPR. Il sert à établir les tables de routage. Aussi, pour qu'il soit diffusé sur tout le réseau, la valeur du TTL dans l'header du message est 255, la valeur maximale.

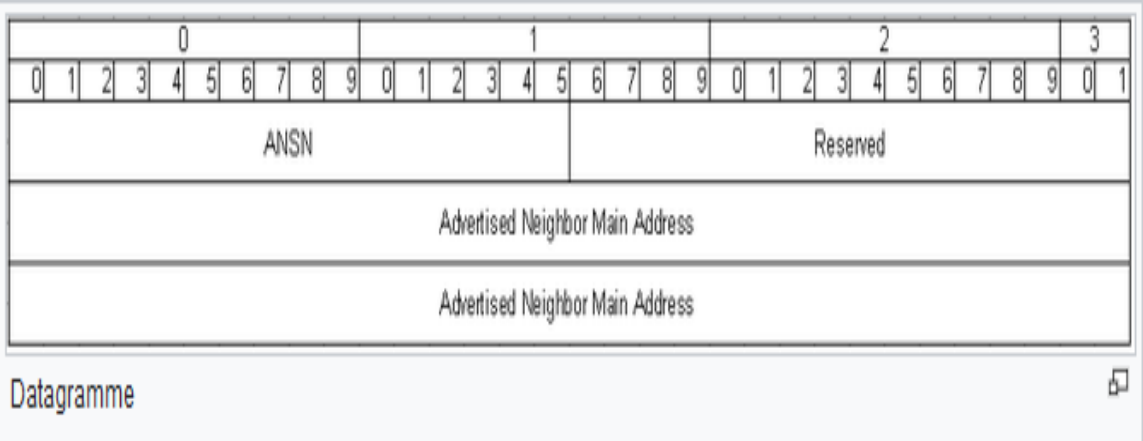


Figure 20 : Format d'un message TC. [26]

- « **Reserved** » : Ce champ doit contenir « 0000000000000000 ».
- « **ANSN** (advertised neighbor sequence number) » : entier incrémenté à chaque changement de topologie. C'est un nombre de séquence du voisin pour maintenir l'information la plus récentes.
- « **Advertised neighbor main address** » : adresse IP de nœuds à un saut. L'ensemble des nœuds publiés dans les messages TC est un sous-ensemble des voisins à un saut. La version par défaut recommande de publier les MPR-selectors, c'est-à-dire les voisins pour lesquels le nœud courant est un relai MPR.

3.3.4.3. Message MID (*multiple interface déclaration*)

Ces messages sont transmis par les nœuds exécutant le protocole OLSR sur plus d'une interface. Les messages MID sont utilisés pour relier les adresses des interfaces OLSR et les adresses principales pour des nœuds OLSR à interfaces multiples.

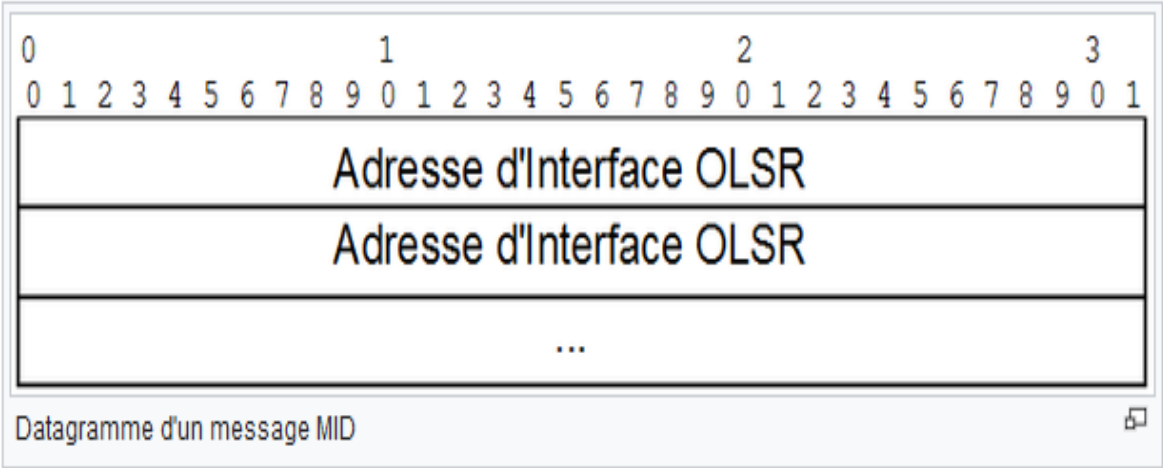


Figure 21 : Format d'un message MID. [26]

Le message MID contient la liste d'adresses des interfaces associées à son adresse principale. Il est envoyé par le nœud dans le réseau pour les déclarer à tous les autres nœuds. Pour obtenir une meilleure fiabilité et un meilleur débit, les messages MID peuvent servir à sélectionner plusieurs interfaces comme principales et ainsi établir des chemins multiples entre deux nœuds voisins. Pour le routage, un nœud à interfaces multiples apparaîtra comme deux nœuds séparés. Si un lien est en panne, un nœud avec de multiples interfaces pourrait encore fournir le chemin de routage pour les autres nœuds.

3.3.4.4. Messages HNA (Host and Network Association)

Ils sont émis par les nœuds ayant des interfaces non-MANET multiples, dont le but est de fournir la connectivité d'un réseau OLSR à un réseau non OLSR. Le nœud passerelle émet des messages HNA contenant une liste d'adresses des réseaux associés et leurs masques réseau (Netmasks). Donc, les nœuds se trouvant dans les réseaux MANET vont construire des tuples pour tous les nœuds passerelles où chaque tuple contient :

- **Geteway_addr** : adresse principale du nœud passerelle.
- **Network_addr** : adresse de sous réseau.
- **Netmask** : adresse de masque réseau.
- **Time** : la durée de vie du tuple.

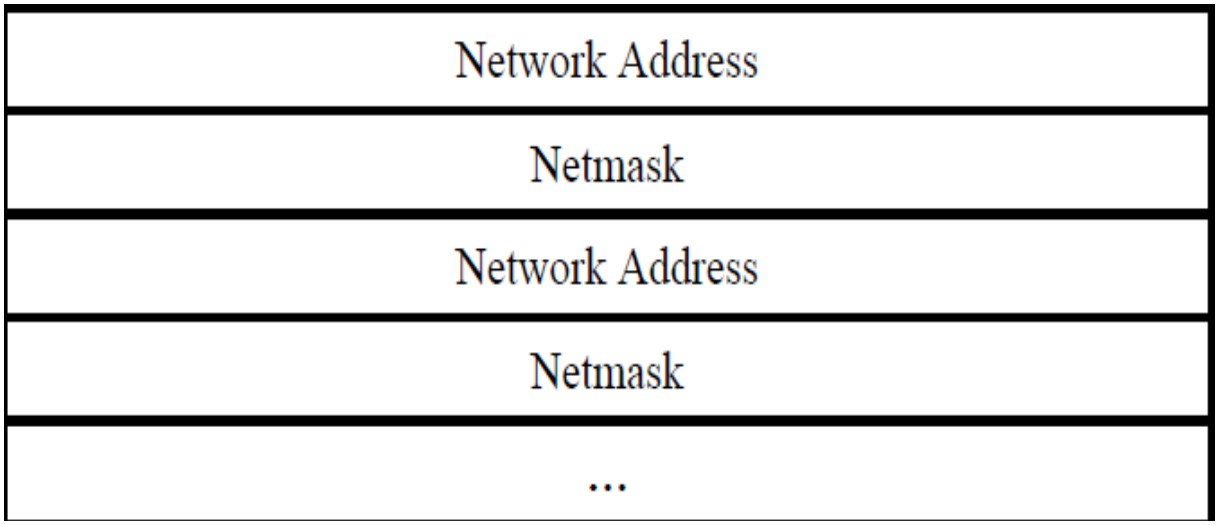


Figure 22 : Format d'un message HNA. [26]

3.4. Référentiels d'informations

Par l'échange de messages de contrôle OLSR, chaque nœud accumule informations sur le réseau. Ces informations sont stockées selon aux descriptions de cette section **[24]**.

3.4.1. Base d'informations sur les associations d'interfaces multiples

Pour chaque destination du réseau, "Interface Association Tuples" (I_iface_addr, I_main_addr, I_time) sont enregistrés. I_iface_addr est une adresse d'interface d'un nœud, I_main_addr est l'adresse principale de ce nœud. I_time spécifie l'heure à laquelle ce tuple expire et doit être supprimé.

Dans un nœud, l'ensemble des tuples d'association d'interface est noté "Ensemble d'association d'interface".

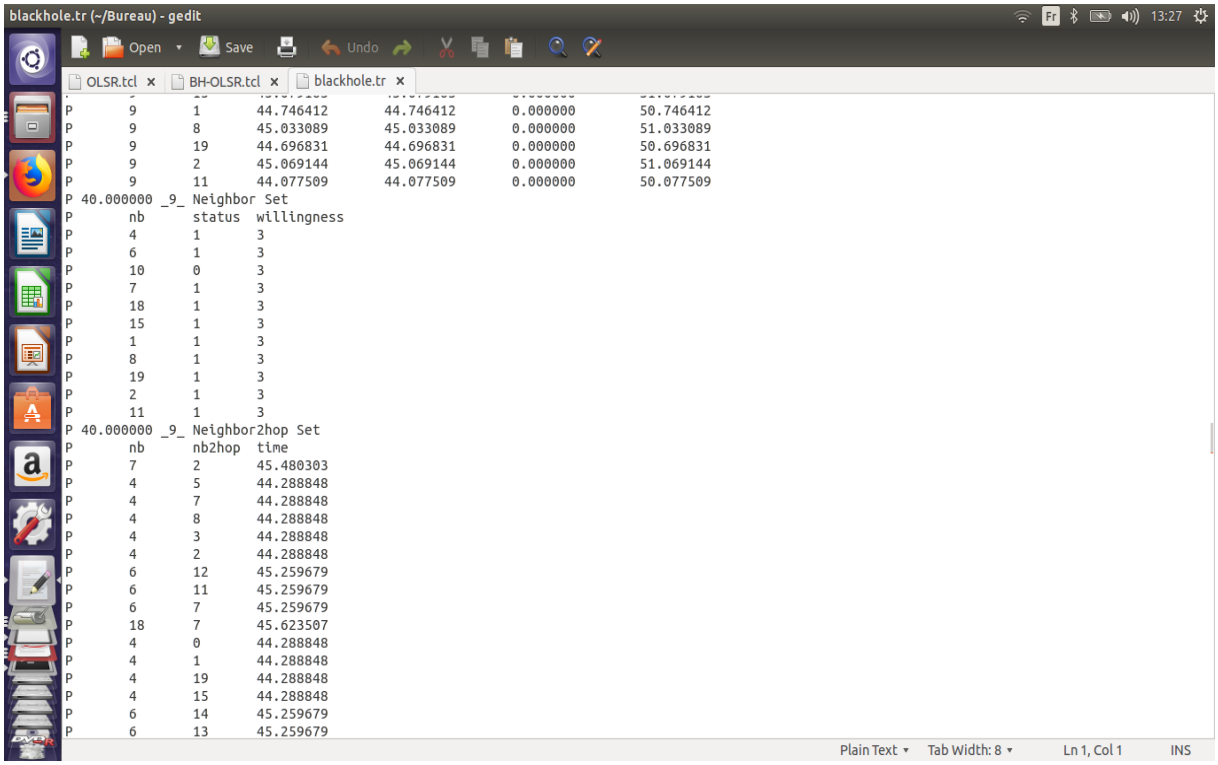


Figure 23 : Exemple de base d'informations.

3.4.2. Détection de lien : base d'informations de lien local

La base d'informations sur les liens locaux stocke des informations sur les liens vers voisins.

3.4.2.1. Ensemble de liens « Link set »

Un nœud enregistre un ensemble de "Link Tuples" (L_local_iface_addr, L_nequart_iface_addr, L_SYM_time, L_ASYM_time, L_time). L_local_iface_addr est l'adresse d'interface du nœud local (i.e., une extrémité du lien), L_nequart_iface_addr est l'interface l'adresse du nœud voisin (c'est-à-dire l'autre extrémité du lien), L_SYM_time est le temps jusqu'à lequel le lien est considéré comme symétrique, L_ASYM_time est l'heure jusqu'à laquelle l'interface voisine est considéré comme entendu, et L_time spécifie l'heure à laquelle cet enregistrement expire et doit être supprimé. Lorsque L_SYM_time et L_ASYM_time sont expiré, le lien est considéré comme perdu. **[24]**

Ces informations sont utilisées lors de la déclaration des interfaces voisines dans les messages HELLO.

L_SYM_time est utilisé pour décider du type de lien déclaré pour le voisin interface. Si L_SYM_time n'est pas expiré, le lien doit être déclaré symétrique. Si L_SYM_time est expiré, le lien doit être déclaré asymétrique. Si L_SYM_time et L_ASYM_time sont tous deux expirés, le lien doit être déclaré perdu. Dans un nœud, l'ensemble des tuples de lien est désigné par "Link set ».

3.4.3. Détection de voisin : base d'informations de voisinage

La base d'informations de voisinage stocke des informations sur les voisins, Voisins à 2 sauts, MPR et sélecteurs MPR.

3.4.3.1. Ensemble de voisin « Neighbor set »

Un nœud enregistre un ensemble de "tuples voisins" (N_nequart_main_addr, N_status, N_willingness), décrivant les voisins. N_nequart_main_addr est l'adresse principale d'un voisin, N_status spécifie si le nœud est NOT_SYM ou SYM. N_willingness dans un entier entre 0 et 7, et spécifie la volonté du nœud de transporter le trafic pour le compte d'autres nœuds.
Dans un nœud, l'ensemble des tuples à un seul saut est désigné par le " Neighbor set ".

3.4.3.2. Ensemble de voisin à 2 sauts « Neighbor 2hop set »

Un nœud enregistre un ensemble de "tuples à 2 sauts" (N_nequart_main_addr, N_2hop_addr, N_time), décrivant symétrique (et, puisque MPR lie par définition sont également symétriques, donc également des liens MPR) entre ses voisins et le voisinage symétrique à 2 sauts. N_nequart_main_addr est l'adresse principale d'un voisin, N_2hop_addr est l'adresse principale d'un voisin à 2 sauts avec un lien symétrique vers N_nequart_main_addr, et N_time spécifie l'heure à laquelle le tuple expire et doit être supprimé. **[24]**

Dans un nœud, l'ensemble des tuples à 2 sauts est désigné par le " Neighbor 2hop set ".

3.4.3.3. Ensemble des MPRs

Un nœud maintient un ensemble de voisins qui sont sélectionnés comme MPR. Leur les adresses principales sont répertoriées dans l'ensemble MPR.

3.4.3.4. Ensemble des sélecteurs MPR

Un nœud enregistre un ensemble de tuples MPR-selector (MS_main_addr, MS_time), décrivant les voisins qui ont sélectionné ce nœud comme MPR. MS_main_addr est l'adresse principale d'un nœud, qui a sélectionné ce nœud comme MPR. MS_time spécifie l'heure à laquelle le tuple expire et doit être supprimé.

Dans un nœud, l'ensemble des tuples de sélecteur MPR est noté "MPR selctor set".

3.4.4. Base d'informations sur la topologie « Topologie Set »

Chaque nœud du réseau conserve les informations de topologie sur le réseau. Ces informations sont acquises à partir des messages TC et sont utilisées pour les calculs de table de routage. Ainsi, pour chaque destination du réseau, au moins une Topologie, Le tuple (T_dest_addr, T_last_addr, T_seq, T_time) est enregistré. T_dest_addr est l'adresse principale d'un nœud, qui peut être atteinte dans un saut du nœud avec l'adresse principale T_last_addr. Typiquement, T_last_addr est un MPR de T_dest_addr. T_seq est un numéro de séquence, et T_time spécifie l'heure à laquelle ce tuple expire et doit être supprimé.

Dans un nœud, l'ensemble des tuples de topologie est dénommé « Topologie set ».

3.4.5. Calcul de la table de routage

Chaque nœud maintient une table de routage qui lui permet d'acheminer les données, destiné aux autres nœuds du réseau. La table de routage est construite sur la base des informations contenues dans la base d'informations de liaison locale et l'ensemble de topologie. Par conséquent, si l'un de ces ensembles est modifié, la table de routage est recalculée pour mettre à jour les informations de route à propos de chaque destination du réseau. Les entrées d'itinéraire sont enregistrées dans la table de routage au format suivant :

1. R_dest_addr R_next_addr R_dist R_iface_addr
2. R_dest_addr R_next_addr R_dist R_iface_addr

Chaque entrée de la table se compose de R_dest_addr, R_next_addr, R_dist, et R_iface_addr. Une telle entrée spécifie que le nœud identifié par On estime que R_dest_addr correspond à des sauts de R_dist loin du nœud local, que le nœud voisin symétrique avec l'adresse d'interface R_next_addr est le nœud de saut suivant dans la route vers R_dest_addr, et que ce le nœud voisin symétrique est accessible via l'interface locale avec l'adresse R_iface_addr. Les entrées sont enregistrées dans la table de routage pour chaque destination du réseau pour laquelle un itinéraire est connu.

Toutes les destinations pour lesquelles un itinéraire est interrompu ou seulement partiellement connus, ne sont pas enregistrés dans le tableau. Plus précisément, la table de routage est mise à jour lorsqu'un changement est détecté soit dans :

- L'ensemble de liens.
- L'ensemble voisin.
- L'ensemble voisin à 2 sauts.
- L'ensemble de topologie.
- La base d'informations sur les associations d'interfaces multiples.

3.5. Les failles de sécurité dans OLSR

- Problème de synchronisation des messages TC et des informations contenues dans chaque nœud.
- La localisation des nœuds
- L'absence d'un point fixe dans le réseau.
- La négligence de la qualité des liens interconnectant.
- Consommation d'énergie.

Pour faire face aux failles de sécurité du protocole OLSR, différents protocoles sont en cours de développement ou ont été développés afin de répondre aux carences du protocole.

- **Batman :**

OLSR souffre d'un sérieux problème de synchronisation des messages TC et des informations relatives aux routes contenues dans chacun des nœuds. En effet les routes connues par les nœuds et l'état réel de la topologie peuvent différer durant un certain laps de temps, et il est nécessaire d'attendre que la mise à jour de la topologie soit effectuée afin de disposer de routes à nouveau correctes, ce qui peut créer des boucles au sein du réseau.
C'est une des raisons pour lesquelles le développement du protocole BATMAN fut lancé.

- **CE-OLSR :**

CE-OLSR est une version basée sur OLSR mais intégrant la localisation des nœuds. Chaque message HELLO se voit ainsi ajouter un champ dans lequel sa propre position (coordonnées GPS) ainsi que celles de ses voisins sont ajoutées. Chaque message TC quant à lui se voit également ajouter par les MPR un champ dédié aux coordonnées des autres sélecteurs MPR.
CE-OLSR apporte ainsi les améliorations suivantes :

- Meilleure robustesse des messages de contrôle (topologie) par rapport à la perte de paquets.
- Les changements de topologie sont détectés plus rapidement, ce qui apporte un suivi précis du voisinage.
- Capacité à discerner les informations cartographiques récentes des obsolètes.
- Il en résulte de meilleures performances, sur le plan à la fois du débit et du temps de réponse et également une validité accrue des routes connues.

- **M-OLSR :**

M-OLSR est une variante d'OLSR modifiant le protocole afin de fournir de meilleures performances au sein de WMN (réseaux maillés sans-fil), à savoir :

- Amélioration du débit.
- Meilleur pourcentage de paquets délivrés.
- Réduction de la surcharge réseau liée au protocole.

Les WMN diffèrent des MANET par leur architecture quelque peu différente. En effet, alors qu'un MANET ne possède aucun point fixe parmi tous ses nœuds, un WMN possède un nœud connu de tous les autres, jouant le rôle de colonne dorsale (backbone) au sein du réseau. La répercussion d'une modification du protocole OLSR n'est donc pas forcément la même selon que le réseau concerné est de type WMN ou MANET.

- **Sélection des MPR :**

Une première manière de procéder consiste à modifier l'algorithme de sélection des MPR. Le comportement défini par la RFC d'OLSR indique que cette sélection s'effectue sur base du nombre de voisins, le nœud élu en tant que MPR étant celui possédant le plus de voisins à 2 sauts. Cette méthode de sélection ne prend cependant pas en compte la qualité des liens interconnectant les nœuds, ce qui dans une optique de qualité de service est pourtant un des points les plus critiques.

Un autre algorithme pourrait consister à se baser sur la bande passante disponible en tant que premier critère de sélection des MPR. Le but est alors d'obtenir une route possédant un débit le plus élevé possible, facteur jouant un rôle prépondérant dans les applications temps-réel de plus en plus utilisées aujourd'hui.

À l'origine OLSR sélectionne le chemin le plus court lors de la sélection des MPR, ce qui n'est pas forcément synonyme de performance, encore faut-il définir le terme "performance" (latence, bande-passante, redondance, etc.). **[27]**

Dans le cas où l'on souhaite orienter les performances vers un aspect ou un autre en particulier, il faut modifier l'algorithme afin de sélectionner les MPR selon d'autres facteurs que la connectivité des nœuds, comme la vitesse, dans le cas où la bande passante est la contrainte première du QoS.

- **Consommation énergétique :**

Les nœuds du réseau MANET ne sont pas connectés sur le réseau électrique et il en résulte que l'énergie de leur batterie est limitée. La vie de la batterie peut aussi affecter la performance de communication du réseau. Quand un nœud épuise l'énergie disponible, il cesse de fonctionner et il peut y avoir pénurie d'hôtes mobiles dans le partitionnement du réseau. Pour cette raison la réduction de la consommation énergétique est un sujet important dans les réseaux sans fil ad hoc. Le protocole OLSR ne tient pas compte de la consommation énergétique pendant l'élection des nœuds MPR, ni pendant le routage des paquets de données et ne tire aucun avantage à partir de l'information des liaisons unicast du réseau.

Les évolutions envisagées pour prendre en compte ces aspects sont entre autres les protocoles EE-OLSR, EOLSR et DE-OLSR.

3.6. Type d'attaques contre OLSR

On peut répertorier les attaques en deux catégories :

- Attaques communes à tous les protocoles de routage proactifs.
- Attaques inhérentes au protocole OLSR.

Ce mémoire portant uniquement sur le protocole OLSR, les vulnérabilités présentées concernent majoritairement les attaques propres à ce protocole. Chaque nœud a pour rôle premier de générer du trafic propre au protocole de routage, et deuxièmement est responsable de relayer les informations de routage des autres nœuds du réseau. Un comportement incorrect résulte donc de la génération d'informations erronées sur le routage, ou de l'absence de relais de ces informations. Une attaque permettant de fournir une connectivité illicite au réseau doit résulter d'un comportement anormal d'au moins un des nœuds qui le composent.

Une attitude anormale peut résulter de deux comportements :

- Un nœud supposé connecter au réseau a été compromis et ne possède plus les mêmes caractéristiques conformes au protocole qu'auparavant.
- Un nœud fait partie du réseau alors qu'il ne le devrait pas.

Il existe de nombreuses possibilités afin d'introduire des nœuds dans le protocole OLSR (en partant d'un OLSR définit par la RFC, dépourvu de procédure de validation) pour lancer différentes attaques.

- Une surcharge des routeurs en inondant le réseau afin de le saturer ("déni de service" ou "Denial of service").
- L'envoi de messages de mises à jour invalides.

3.6.1. Brouillage

OLSR est vulnérable au brouillage, c'est d'ailleurs le cas pour tous les autres protocoles de routage utilisés sur réseaux ad-hoc. Le brouillage consiste à générer une grande quantité d'interférences radio. Le bruit généré entraîne alors l'impossibilité pour les nœuds de s'échanger des informations utiles entre eux, notamment leurs routes respectives, et empêche ainsi la construction d'un réseau. Cette vulnérabilité ne peut pas être corrigée au niveau du protocole de routage.

3.6.2. Envoi des messages de mises à jour invalides

Un nœud a normalement deux responsabilités :

- Générer des messages de contrôles.
- Les transférer.

Pour compromettre l'intégralité du protocole de routage, l'attaquant peut soit envoyer des paquets de contrôle incorrects lorsque le nœud génère les messages de contrôle, soit les modifier lorsque les messages de contrôle sont envoyés. Un message de contrôle peut être trafiqué en changeant son identité (en anglais "identity spoofing") ou en endommageant ses données (en anglais "Link spoofing").

3.6.3. Attaque sur le message de contrôle durant sa génération

Dans les schémas suivants, les nœuds jaunes sont des nœuds classiques et les autres des MPR.

3.6.3.1. Usurpation d'identité avec un message HELLO

Un nœud malicieux prétend en être un autre en usurpant son adresse IP (Usurpation d'adresse IP), afin d'envoyer un message HELLO à son voisinage.

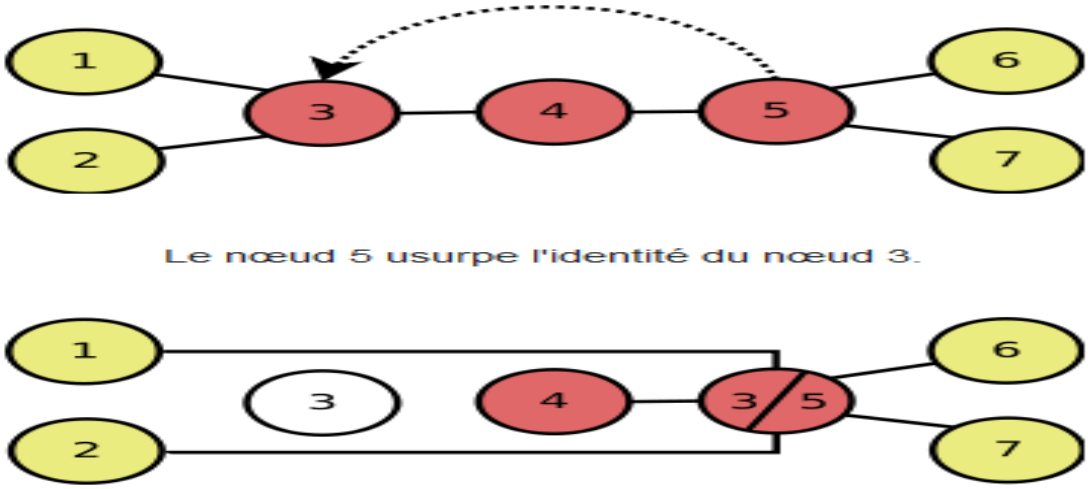


Figure 24 : Usurpation d'identité d'un nœud avec message HELLO. [26]

Les nœuds 1 et 2 sont persuadés d'être connectés au nœud 3 mais en réalité le sont au nœud 5, qui se fait passer pour le nœud 3.

3.6.3.2. Corruption des données du message HELLO

Les messages HELLO vont contenir des falsifications de la topologie du réseau avec l'insertion de nœuds inexistants, pour que des usurpateurs soient reconnus comme des MPR, pouvant alors contrôler tous les flux qui passent par eux. Il est aussi possible de supprimer des nœuds existants (par omission dans les messages HELLO) afin de les rendre inaccessibles dans la topologie ainsi construite.

3.6.3.3. Usurpation d'identité d'un nœud avec un message TC

Le nœud 5 est censé envoyer un message TC indiquant qu'il est le dernier saut jusqu'aux nœuds 6 et 7.

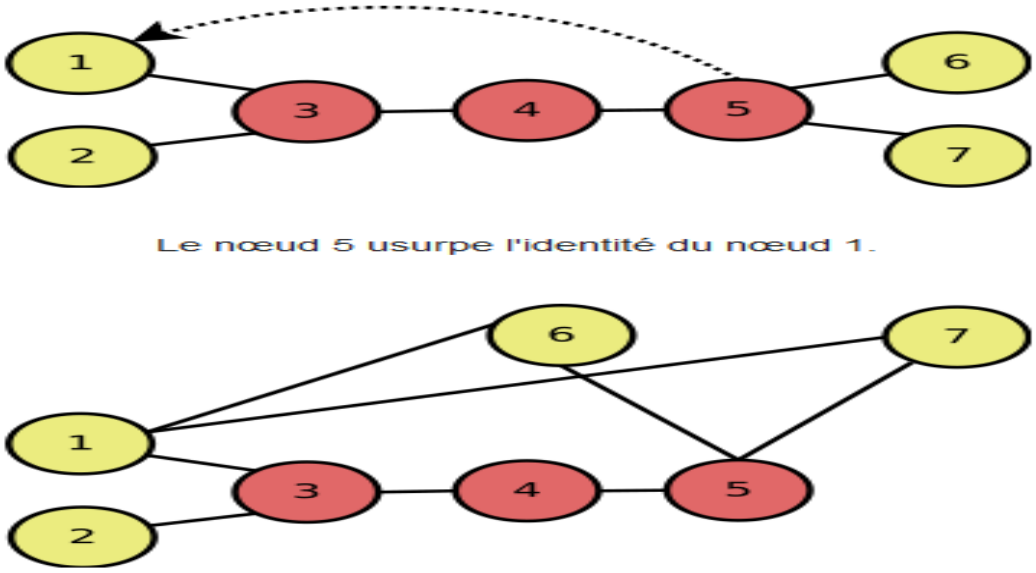


Figure 25 : Usurpation d'identité d'un nœud avec message TC. [26]

3.6.3.4. Corruption des données avec un message TC

La corruption des données dans les messages TC consiste à insérer des MPR inexistants ou à en supprimer. La suppression de MPR au sein des tables de routage risque de fragmenter le réseau et certains MPR ne seront plus accessibles. L'insertion de nœuds MPR non existants permet de créer de faux liens et de déformer la topologie du réseau, ce qui aura pour conséquence de créer des "boucles" lors du routage ou de générer des conflits lors du calcul des tables de routage.

3.6.4. Attaque sur le message de contrôle durant la transmission

Les messages TC nécessitent d'être transmis à travers les MPR à l'ensemble du réseau, afin de transporter des informations critiques sur les tables de routage. Les nœuds relais peuvent trafiquer ces messages pendant leur transfert, en ajoutant ou supprimant des MPR. Les problèmes peuvent être plus sérieux que lors du trafic des messages HELLO, car les messages TC sont utilisés par tous les nœuds. Il est donc très facile de lancer une attaque de type "trou noir", qui met en œuvre des nœuds qui font discrètement disparaître le trafic.

3.7. Mécanismes de sécurité proposés pour OLSR

Ces dernières années, de nombreuses contributions ont été proposées pour la sécurité du protocole OLSR. Dans cette section, nous présenterons une revue des principales solutions proposées.

3.7.1. Solutions utilisant la cryptographie asymétrique

Dans la littérature, plusieurs extensions de sécurité pour le protocole OLSR basées sur la cryptographie ont été proposées. Leur point commun réside dans l'utilisation de signature numérique pour assurer l'authentification et intrinsèquement l'intégrité des messages de contrôle.

3.7.1.1. Secure OLSR

Dans **Secure OLSR**, Hafslund et al. Ont proposé une approche d'authentification de saut en saut dans laquelle chaque nœud signe les paquets OLSR au fur et à mesure de leur retransmission. La signature numérique est calculée sur le corps et l'entête du message, en utilisant la cryptographie asymétrique. Un message SIGNATURE est généré et envoyé avec tout autre message de contrôle (HELLO, TC, MID ou HNA). Ainsi, à la réception d'un paquet OLSR, un nœud intermédiaire vérifie la signature du nœud précèdent, la retire, puis appose sa propre signature. Cette approche permet d'inclure dans le calcul de la signature numérique les champs devant être modifiés en transit, tels que le TTL (Time To Live) et le nombre de sauts. Cependant, la signature assure seulement que le nœud qui a transmis le trafic est bien celui qui a signé le paquet, mais n'apporte aucune garantie sur l'intégrité du paquet original. De manière similaire à Secure OLSR, Raffo et al. ont proposé une extension de sécurité pour le protocole OLSR basée sur l'utilisation de signatures numériques. Une première différence se situe au niveau du type des données protégées. Dans leur approche, une signature numérique est associée à chaque message de contrôle OLSR (i.e. HELLO ou TC) et non plus à chaque paquet OLSR. Ensuite, les auteurs proposent une approche d'authentification de bout en bout selon laquelle un nœud récepteur d'un message de contrôle authentifie le nœud d'origine plutôt qu'un nœud intermédiaire dans son cheminement. Ici, les champs TTL et nombre de sauts ne sont pas protégés par la signature, car ces derniers doivent être modifiés en transit par chaque nœud intermédiaire. La solution pour pallier le problème et ainsi faire face aux attaques par retransmission des messages de contrôle (*Replay Attacks*), les auteurs proposent d'horodater chaque message OLSR, en utilisant un *Timestamp* dans les messages au lieu du TTL. Le *Timestamp* est inséré lors de la création du message en même temps que la signature. Ainsi, lorsqu'un nœud reçoit un message de contrôle, il contrôle le *Timestamp* et vérifie la signature du message. Si le *Timestamp* et la signature sont corrects, le noeud traite le message, sinon ce dernier le rejette, alors le nœud malicieux, originaire de ce message, se trouve isolé du reste du réseau. L'authentification des messages de contrôle représente une première ligne de défense pour contrecarrer efficacement les attaques externes contre le protocole OLSR. Or à elle seule, l'authentification n'empêche pas l'inoculation de fausses informations de routage par des attaquants internes au réseau. Pour pallier ce problème, d'autres méthodes plus originales ont été proposées. Parmi ces méthodes, on trouve la méthode **ADVSIG**. **[27]**

3.7.1.2. An Advanced Signature System for OLSR (ADVSIG pour OLSR)

Les mécanismes de signature et de Timestamp ne sont pas suffisants pour faire face aux différents types d'attaques. En effet, cette solution n'est pas efficace dans le cas où un noeud légitime est compromis car ce noeud malicieux peut alors générer des messages signés correctement avec son identité et ainsi envoyer de faux messages de contrôle à travers le réseau. Dans ce contexte, un mécanisme additionnel ADVSIG (Advanced Signature) a été proposé par Raffo et al. Le but du schéma ADVSIG est d'empêcher l'injection d'informations d'état de liens non valides par des nœuds attaquants internes, d'assurer l'intégrité du réseau et potentiellement éviter les nœuds malveillants dans la phase d'établissement des routes. Le schéma proposé s'appuie sur le fait que la topologie du réseau évolue selon une séquence chronologique précise et en particulier, que l'état de lien entre deux nœuds au temps (t + 1) dépend directement de l'état de ce même lien au temps (t). L'idée est que pour tout message reçu d'un voisin, un nœud stocke les informations relatives à ses liens, puis les réutilise en tant que preuve de validité de son ensemble d'état de liens dans les messages qu'il émet ultérieurement. Pour éviter toute fabrication de faux messages, ces informations sont signées et encapsulées dans un message spécifique nommé ADVSIG généré et envoyé avec les messages de contrôle HELLO et TC. Ainsi, un message de contrôle envoyé par un nœud compromis ne pourra donc contenir de faux liens, parce que ces liens manquent de preuves appropriées. Le message ADVSIG contient aussi un champ estampille temporelle, obtenue de l'horloge interne du nœud, pour éviter les attaques de rejeux. Cette solution assure une bonne authentification des messages de contrôle ainsi qu'une bonne intégrité et ne demande pas de modification des messages standards du protocole OLSR. Cependant, des surcoûts induits par les opérations de calculs et de vérifications des signatures numériques conduisent à des pertes notables de messages.

En plus, ce mécanisme ne fournit pas une solution pour les attaques de type DoS ou wormhole et il ajoute une charge de trafic importante à cause des signatures échangées entre les nœuds.**[28]**

3.7.1.3. Approche basée sur la corrélation des informations contenues dans les messages HELLO et TC

Dans cette catégorie de mécanisme de contre-mesure, aucune modification du protocole n'est requise. L'idée proposée initialement par Wang et al. et puis reprise par Cuppens est de dériver des propriétés de sécurité pour le protocole OLSR à partir de la corrélation des informations contenues dans les messages HELLO et TC. Les relations/règles qui doivent être vérifiées pour qu'une information soit considérée comme valide sont les suivantes :

- La relation HELLO-TC définit que chaque noeud annoncé MPR selector dans un message TC doit également être annoncé comme étant un voisin symétrique du noeud originaire dans un message HELLO antérieur.
- La relation MPR-MPR définit qu'un noeud annoncé MPR selector dans un message TC doit avoir annoncé le noeud originaire du message TC dans le MPR_set d'un message HELLO antérieur.
- La relation intégrité des messages définit que dans la phase de retransmission des messages TC, l'en-tête OLSR ne doit subir aucune modification.
- La relation voisinage-MPR définit qu'un noeud générateur d'un message TC doit être annoncé dans les messages HELLO de chaque noeud déclaré MPR Selector comme étant un de leur voisin symétrique.

- Avantages : L’approche ne requiert pas de modification du format des messages du protocole. Il s'agit d'un raisonnement sur le protocole OLSR formalisation des relations extraites/dérivé à partir de la corrélation des informations contenues dans les messages HELLO et dans les messages TC.
- Inconvénients : Néanmoins, la validation de l'approche n'est pas formelle, l’efficacité de l'approche est vérifiée seulement par l'expérimentation. La dernière relation requiert que chaque noeud puisse observer les retransmissions de ses voisins. Une autre limite de cette approche est que les coalitions d'adversaire ne sont pas étudiées.

3.7.1.4. Mécanisme de réputation basé sur la contreréaction

Une approche de réputation basée sur la contre-réaction pour faire face aux attaques de type mystification de lien a été proposée par Vilela et Barros. L'objectif de ce mécanisme est de s'assurer que chaque nœud génère correctement les messages de contrôle. Pour ce faire, deux opérations sont utilisées : le message de contre-réaction (message de feedback) et la table d'évaluation. Le message de contre-réaction est utilisé pour indiquer le chemin parcouru par le message de contrôle TC. Quand un nœud MPR reçoit un message TC, il envoie, au nœud originaire du message TC, un message de contre-réaction indiquant le chemin traversé par le message de contrôle. D'autre part, une table d'évaluation est maintenue par chaque nœud dans le réseau afin d'évaluer et classifier les autres nœuds selon leur réputation de transmission correcte des messages de contrôle. La réputation est générée grâce à un mécanisme de surveillance WATCHDOG. A la réception du message de contre-réaction, les nœuds qui se conduisent mal (qui génèrent de fausses informations de routage) sont pénalisés par la diminution de leur réputation, inversement la réputation des nœuds qui se comportent bien est augmentée. Cependant, ce mécanisme se base sur la détection distribuée et ne tient pas compte du cas de mauvais témoignages. Un nœud malveillant qui se comporte correctement peut ainsi obtenir une réputation positive, pour ensuite fournir une mauvaise réputation sur d'autres nœuds légitimes se conduisant correctement afin de les pénaliser auprès des autres nœuds du réseau. **[30]**

3.7.1.5. WATCHMAN (An Overlay Distributed AAA Architecture for Mobile Ad hoc Networks)

Cette solution se base sur une architecture AAA (Authentication Authorization, Accounting) distribuée et hiérarchique appelée WATCHMAN. Dans cette architecture chaque nœud est identifié par sa paire de clés (privée et publique), et dispose d'une autorisation de niveau de confiance (ATL : Authorization Trust Level) qui lui permet d'accéder aux différentes ressources du réseau. La solution se base sur deux types de serveurs : maître et esclave. Le serveur maître dispose des données d'authentification et d'autorisation et il effectue l'initialisation du réseau. Cette sélection se fait en basant sur différents critères : niveau de confiance, ressource, position dans le réseau. Le serveur esclave dispose des informations nécessaires tel que : la liste des autres serveurs, leurs clés publiques, leurs autorisations et leurs niveaux de confiance…etc. Ce qui lui permet de remplacer le serveur maître dans le cas où ce dernier est indisponible (le serveur esclave qui a les meilleurs critères remplace le serveur maître).

Les auteurs de WATCHMAN proposent d'utiliser le niveau de confiance ATL comme critère de sélection des MPRs, ce qui diminue les problèmes de retransmission de trafic incorrecte et la génération des messages TC incorrectes. Cependant, la solution nécessite l'existence d'une autorité centralisée car tous les nœuds doivent être authentifiés avant de participer aux activités du réseau, ce qui rend la solution incompatible avec la philosophie des réseaux Ad hoc. **[29]**

3.7.1.6. Mécanismes de contre mesure pour l'attaque du trou de ver

La particularité de l'attaque du trou de ver est qu'elle ne nécessite aucune modification des messages de la part de l'adversaire. La conséquence directe est qu'aussi bien les nœuds légitimes (c'est à dire les nœuds en possession des autorisations nécessaires pour participer aux opérations réseau) que les nœuds illégitimes peuvent la mettre en œuvre. Par conséquent, bien que l'ensemble des mécanismes de sécurisation des protocoles de routage basés sur des méthodes cryptographiques offrent des garanties de sécurité en termes de confidentialité, d'authenticité, et d'intégrité des messages, ils ne sont pas résistants à ce type d'attaque. Plusieurs moyens ont été actuellement proposés dans la littérature pour conter cette attaque. Parmi ces moyens, on trouve les mécanismes utilisés au niveau de la couche de routage.

3.7.1.7. Packet Leashes

Packet leashes est une solution de détection de l'attaque du trou de ver proposée par Hu et al. Un leash est une information de temps ou de positionnement géographique qui est incluse dans chacun des paquets émis sur le réseau et qui sert à restreindre leur distance maximale de transmission autorisée. Deux méthodes d'utilisation des leashes sont présentées : une première basée sur le support d'un service de positionnement géographique et une seconde basée sur une synchronisation d'horloge précise (fine) entre les noeuds.

- **Leashes géographiques :** Les leashes géographiques permettent d'assurer la distance entre le récepteur et l'émetteur d'un message. Le mécanisme requiert d'une part que chaque noeud connaisse sa propre position géographique, et d'autre part que les horloges de tous les noeuds soient lâchement synchronisées (de l'ordre de la milliseconde). A l'émission d'un message, le noeud émetteur inclut dans le message une version authentique de sa propre position géographique et l'heure d'émission. Un noeud récepteur utilise les informations de leashes encapsulées dans le message reçu ainsi que sa propre position géographique et l'heure de réception du message enregistrée pour estimer une borne supérieure de la distance avec l'émetteur. En prenant en considération certaines variables telles que la vélocité maximale des noeuds, l'erreur maximale dans

le système de synchronisation d'horloge, et l'erreur maximale possible dans le système de positionnement géographique, la borne supérieure de la distance entre l'émetteur et le récepteur peut alors être déterminée. Dans le cas où la distance calculée est supérieure à la portée maximale de transmission, alors le lien est probablement faux. Une des limitations de cette méthode vient du fait qu'elle repose sur un système de positionnement géographique. En effet, la technologie GPS est actuellement inopérante dans les environnements clos (tels que les immeubles), les environnements sous-marins, les environnements soumis à un fort rayonnement magnétique, etc. Se pose également la question de la précision des informations de positionnement fournies par la technologie GPS. Les auteurs précisent que selon l'état de l'art dans la technologie GPS, il est possible d'atteindre une précision d'environ 3 m.

- **Leashes temporelles :**

  Les leashes temporels assurent que chaque message transmis à travers le réseau encapsule une borne supérieure sur sa durée de vie. Un paquet reste valide sur le réseau tant que le temps d'expiration n'est pas dépassé, après quoi le paquet est rejeté. Un des prérequis non-négligeable de la méthode est une synchronisation d'horloge précise entre tous les noeuds du réseau. Selon cette méthode, un émetteur inclut dans chaque message une version authentique de l'heure d'émission. Dans la phase de vérification, un récepteur compare cette valeur à l'heure de réception du message. Dans une variante des leashes temporels, un émetteur détermine le temps d'expiration à partir duquel un message ne doit plus être accepté, puis inclus cette information dans le leash. En résumé, la méthode s'appuie sur le temps de parcours d'un message puis sur la vitesse de la lumière pour déterminer sa distance approximative de parcours. Une hypothèse implicite est que les délais de traitement des messages, d'émission et de réception sont négligeables **[31]**.

Solution de sécurité	Principe
Secure OLSR Par « Hafslmid et al » 2004	Une approche d'authentification de saut en saut avec une signature numérique calculée sur le corps et l'entête du message, en utilisant la cryptographie asymétrique.
ADVSIG Par « Raffo et al » 2004	L'idée est que pour tout message reçu d'un voisin, un nœud stocke les informations relatives à ses liens, puis les réutilise en tant que preuve de validité de son ensemble d'état de liens dans les messages qu'il émet ultérieurement.
WATCHMAN Par « R. Khakpour, M Laurent-Maknavicius, and H. Chaouchi » 2008	Dans cette architecture chaque nœud est identifié par sa paire de clés (privée et publique). La solution se base sur deux types de serveurs : maître et esclave. Le serveur maître dispose des données d'authentification et d'autorisation et il effectue l'initialisation du réseau.
Packet leashes Par « Hu et al » 1976_1986.	Un leash est une information de temps ou de positionnement géographique qui est incluse dans chacun des paquets émis sur le réseau et qui sert à restreindre leur distance maximale de transmission autorisée.
Méthodes cryptographiques	Offrent des garanties de sécurité en termes de confidentialité, d'authenticité, et d'intégrité des messages.
Watchdog Et Pathrather Par « Vilela, J. P et J. Barros » 2007	Il détecte les nœuds malveillants en entendant la transmission du prochain saut. Un compteur d'échec est lancé si le nœud suivant ne parvient pas à transmettre le paquet de données. Lorsque la valeur du compteur dépasse un seuil prédéfini, le nœud est marqué comme malveillant.
Approches basées sur les IDS	Est un processus de contrôle et d'analyse des évènements dans un réseau pour détecter et identifier toute tentative d'attaque.

Table 5 : Quelques solutions proposés pour le protocole OLSR.

3.8. Discussion et analyse de notre solution SU-OLSR

Le but de notre approche est de proposer une solution pour faire face aux attaques sur la sélection des MPR par mystification de lien.
Pour se faire, on introduit le concept de confiance entre les noeuds voisins. En effet, chaque noeud ne doit pas faire confiance à un noeud voisin X qui présente des caractéristiques malicieuses et qui pourraient influencer le choix des MPR. On a appelé notre protocole SU-OLSR (SU pour noeuds suspects). Un noeud est dit suspect s'il présente des comportements suspects.

➢ Le nouvel algorithme de sélection des MPR :

Le protocole SU-OLSR utilise un nouvel algorithme de sélection des MPR. L'objectif de cet algorithme est de détecter à la fois l'ensemble des noeuds suspects et aussi l'ensemble des MPR de confiance « vérification ».

L'Algorithme décrit l'heuristique de sélection des MPR pour le protocole SU-OLSR. Pour tout noeud S donné dans le réseau, on commence d'abord par trouver tous ses voisins à 1- saut et à 2-sauts. On recherche par la suite tous les noeuds X dans N1 qui démontrent un comportement suspect. Si c'est le cas, on ajoute les noeuds trouvés à l'ensemble Suspects(S) dans l'ensemble des MPRs. L'étape suivante de l'algorithme est de redéfinir l'ensemble des voisins à 1-saut ainsi que l'ensemble des voisins à 2-sauts basés sur l'ensemble N1
À partir des ensembles N1 et N2, on ajoute à l'ensemble des MPR chaque nœud dans N1 qui couvre un noeud isolé dans N2 et on élimine par la suite ses nœuds isolés de N2 ainsi que les noeuds couverts par l'un des MPR choisi dans cette étape. Tant que tous les noeuds dans N2 ne sont pas tous couverts, on ajoute à l'ensemble des MPR un noeud de N1 qui couvre le maximum de noeuds dans N2. De cette manière, l'ensemble des MPR de confiance du noeud S ainsi que l'ensemble des noeuds suspects sont vérifiés et calculés e selon l'algorithme relatif à SU-OLSR.

L'insertion du l'algorithme SU-OLSR pour détecter l'attaque et faire la vérification si cette dernière existe a été inséré dans :

- ✓ L'ensemble des nœuds a 1 saut les MPR.
- ✓ L'ensemble des nœuds a 2 sauts.
- ✓ L'ensemble des MPRs selectors.
- ✓ Dans le paquet OLSR donc une vérification au niveau de tous les messages d'envoi HELLO,TC,HNA,MID.
- ✓ Au niveau de l'insertion et calcul de la table de routage (ici une vérification générale dans tous les niveaux suivants :
  1. Ensemble d'association d'interface.
  2. Link set.
  3. Neighbor set.
  4. Neighbor 2hop set.
  5. Topologie Set.

➢ Messages de contrôle et algorithme d'inondation dans SU-OLSR :

Une fois que l'ensemble des MPR de confiance a été calculé ainsi que l'ensemble des suspects à travers le réseau, il faut définir un mécanisme pour diffuser les informations de la topologie à travers le réseau.

Le protocole SU-OLSR utilise les mêmes mécanismes d'inondation que le protocole classique OLSR et qui sont donnés dans (Clausen et Jacquet, 2003). Les seuls changements concernent les formats et le contenu du paquet OLSR. On doit modifier les messages de contrôle pour qu'ils prennent en considération les informations concernant le mécanisme de confiance. En effet dans SU-OLSR, il faut que chaque noeud S dans le réseau donne dans ses messages HELLO les MPR de confiance qu'il a choisis ainsi que les nœuds suspects dans ses voisins. De la même manière, il faut que chaque noeud émettant des messages TC déclare dans ses messages les noeuds qui l'ont choisi comme MPR de confiance et aussi les noeuds qui l'ont déclaré comme suspect. Ce changement dans la forme des messages de contrôle n'a aucun impact sur la charge du trafic dans le réseau en comparaison avec le protocole classique OLSR. Du moment où les informations de la topologie sont échangées entre les noeuds selon les nouvelles spécifications des messages de contrôle de SU-OLSR, chaque noeud devrait procéder au calcul des plus courts chemins vers une destination donnée en utilisant l'algorithme de Dijkstra.

➢ Modèle d'attaque BLACKHOLE :

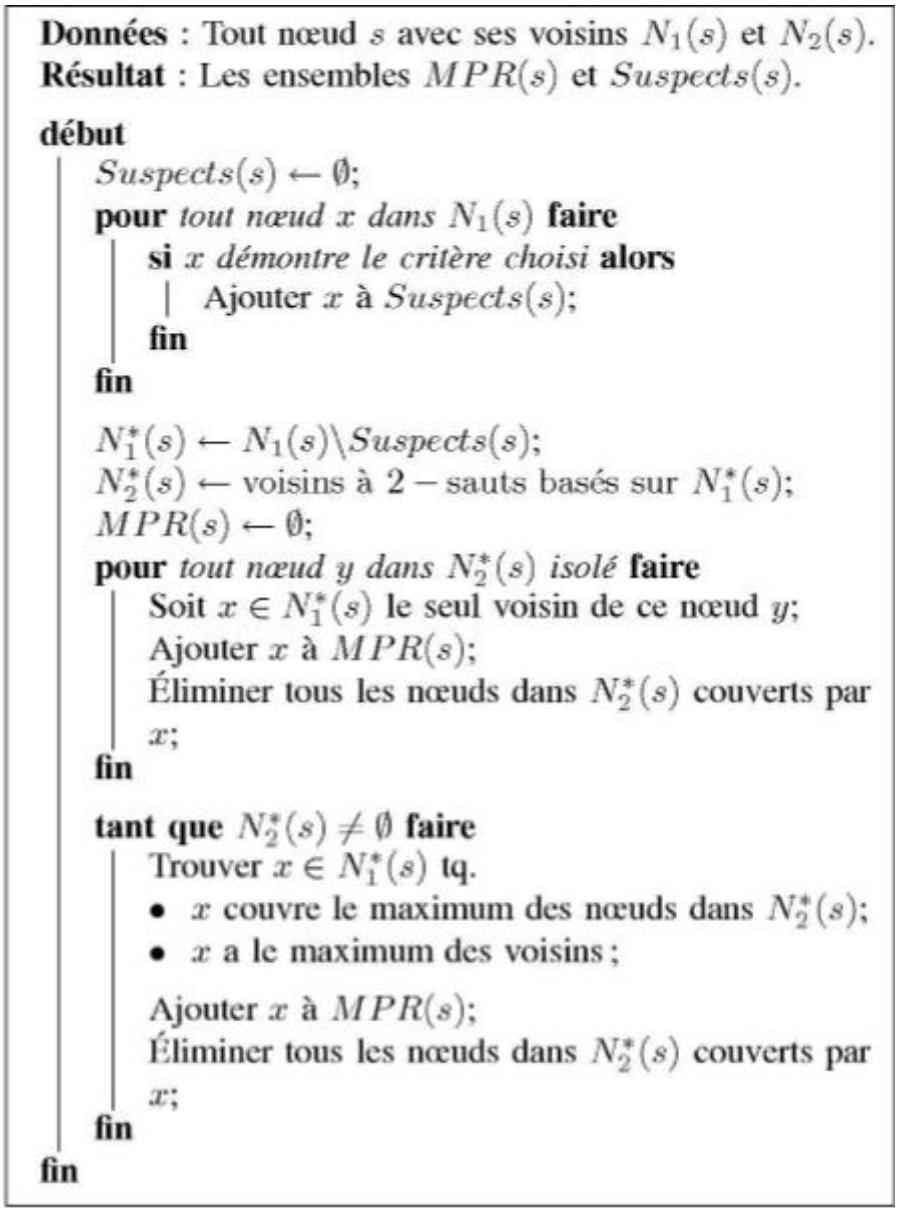
Grâce au protocole SU-OLSR, un noeud malicieux qui déclare qu'il couvre un noeud isolé ou lointain non connu par les autres voisins, ne sera jamais choisi comme MPR. Les seules possibilités qui lui restent sont de mentir sur son réel statut dans le réseau. Afin d'évaluer SU-OLSR à la présence de noeud malicieux, on suppose que les noeuds peuvent mentir sur leur statut dans le réseau.

Comme premier cas, on suppose qu'un noeud malicieux m déclare qu'il a été choisi par un de ses voisins x comme noeud non sécuritaire. Dans ce cas, le nœud malicieux ne bénéficie d'aucune position avantageuse dans le réseau car il est déclaré par les autres noeuds comme non sécuritaire pour relayer leurs messages.

Dans un deuxième cas, on suppose qu'un noeud malicieux m déclare qu'il a été choisi comme MPR de confiance par un noeud x. Dans le cas où x est l'un des voisins de m ou s'il est dans la même partie connexe du graphe du réseau, le noeud x devrait recevoir les messages TC générés par le noeud malicieux et dans lesquels ce dernier déclare que x l'a choisi comme MPR de confiance. À la réception de ces messages TC, le noeud x pourrait initier un mécanisme de contre-mesure pour dénoncer le noeud malicieux auprès des autres noeuds. Le mécanisme de contre-mesure présenté est de détecter le nœud malicieux et de lui rendre un nœud sécuritaire.

D'après la présentation ci-dessus du protocole de routage OLSR et notre solution SU-OLSR, nous remarquons que SU-OLSR offre des fonctionnalités très intéressantes tout en recherchant des routes optimales sécurisé en termes de nombre de sauts. Il diminue au maximum le nombre de messages de contrôle non sécuritaire transmis sur le réseau, en utilisant la technique de sélection des MPR de confiance. SU-OLSR gère convenablement la topologie du réseau, en expédiant périodiquement des messages TC sécurisé.

Tous ces avantages du protocole SU-OLSR ne veut pas dire qu'il n'a pas d'inconvénients, Malgré que ces dernières années beaucoup de recherches ont été faites pour améliorer la protection du OLSR contre les attaques, mais OLSR reste toujours vulnérable à certaines attaques.



Données : Tout nœud s avec ses voisins $N_1(s)$ et $N_2(s)$.
Résultat : Les ensembles $MPR(s)$ et $Suspects(s)$.

début
- $Suspects(s) \leftarrow \emptyset$;
- **pour** *tout nœud* x *dans* $N_1(s)$ **faire**
  - **si** x *démontre le critère choisi* **alors**
    - Ajouter x à $Suspects(s)$;
  - **fin**
- **fin**
- $N_1^*(s) \leftarrow N_1(s) \backslash Suspects(s)$;
- $N_2^*(s) \leftarrow$ voisins à $2-$sauts basés sur $N_1^*(s)$;
- $MPR(s) \leftarrow \emptyset$;
- **pour** *tout nœud* y *dans* $N_2^*(s)$ *isolé* **faire**
  - Soit $x \in N_1^*(s)$ le seul voisin de ce nœud y;
  - Ajouter x à $MPR(s)$;
  - Éliminer tous les nœuds dans $N_2^*(s)$ couverts par x;
- **fin**
- **tant que** $N_2^*(s) \neq \emptyset$ **faire**
  - Trouver $x \in N_1^*(s)$ tq.
    - x couvre le maximum des nœuds dans $N_2^*(s)$;
    - x a le maximum des voisins ;
  - Ajouter x à $MPR(s)$;
  - Éliminer tous les nœuds dans $N_2^*(s)$ couverts par x;
- **fin**

fin

Figure 26 : Sélection des MPRs avec SU-OLSR.

3.9. Conclusion

En conclusion, Dans notre étude du protocole OLSR, on s'est aperçu que son schéma de base est construit sur l'hypothèse d'un environnement idéal dans lequel le fonctionnement du réseau n'est pas soumis à des attaques malveillantes. La réalité contredit cette hypothèse. En effet, il existe plusieurs vulnérabilités permettant à des noeuds malintentionnés de corrompre la configuration des tables de routage, de modifier les paquets en transit ou tout simplement de ne pas participer à l'effort de routage dans le but d'économiser de l'énergie. Ceci est dû au fait qu'un réseau sans fil est vulnérable aux attaques qu'un réseau filaire, car les transmissions radio sont effectuées en l'air. L'intrus peut écouter tous les messages échangés pourvu qu'il se trouve dans l'air d'émission. Contrairement à un réseau filaire où un intrus a besoin d'accéder à une machine ou bien se connecter aux câbles.

Les extensions de sécurité proposées pour OLSR couvrent un grand nombre de problèmes distincts. Une grande partie se base sur des mécanismes cryptographiques pour garantir l'intégrité et authentifier le noeud originaire du trafic de contrôle. L'introduction de *Timestamp* a permis la limitation et la détection des attaques qui utilisent les anciens messages pour les envoyer sur le réseau. Des techniques plus élaborées s'appuient sur la validation de l'information. Il s'agit là d'un niveau avancé de protection qui utilise des connaissances additionnelles, telles que les déclarations précédentes ou bien les données collectées de l'ensemble des participants. Ces techniques restent moins pratiques que les solutions cryptographiques et nécessitent une étude approfondie du comportement des noeuds.
De nombreux travaux de recherche s'ajoutent à ces solutions qui restent des mécanismes partiels. Au jour d'aujourd'hui, il n'existe pas de solution complète faisant face à toutes les vulnérabilités du protocole OLSR.

Le chapitre suivant sera consacré à la présentation de notre solution qui consiste à proposer une approche pour sécuriser les échanges des messages de contrôle dans OLSR.

Chapitre 04

Simulation et discussion des résultats

4.1. Introduction

Dans le chapitre précédent, nous avons présenté une solution de sécurité pour améliorer les performances du protocole OLSR en réduisant le coût calculatoire engendré par les opérations cryptographiques.

Afin d'évaluer les performances de notre solution SU-OLSR par rapport à celle du protocole original OLSR, nous avons effectué des simulations par un simulateur réseau. Nous présentons en premier lieu l'environnement de simulation utilisé avec les métriques de performances mesurées, les scénarios de simulations adoptés, puis nous donnons l'interprétation des résultats obtenus à l'issue de ces simulations.

4.2. Définition de la simulation

Simuler, c'est modéliser un système complexe, afin de prévoir son comportement dans le monde réel. Il s'agit d'une approche permettant de représenter le fonctionnement d'un système réel constitué de plusieurs entités, de modéliser les différentes interactions entre elles, et enfin évaluer le comportement global du système et son évolution dans le temps.

Le recours à la simulation permet de contourner les limites de la complexité des modèles analytiques. Toutefois, il est nécessaire de bien identifier les caractéristiques du système afin de le représenter, le plus finement possible, par des modèles abstraits.

.

4.3. Présentation du simulateur NS2

NS est un outil logiciel de simulation de réseaux informatiques. Il est essentiellement élaboré avec les idées de la conception par objets, de la réutilisation du code et de modularité. Il est aujourd'hui un standard de référence en ce domaine, plusieurs laboratoires de recherche recommandent son utilisation pour tester les nouveaux protocoles.

Le simulateur NS2 actuel est particulièrement bien adapté aux réseaux à commutation de paquets et à la réalisation de simulations de grande taille .NS2 est écrit en C++ et utilise le langage OTCL (Object Tools Command Language) dérivé de TCL. A travers OTCL, l'utilisateur décrit les conditions de la simulation : la topologie du réseau, les caractéristiques des liens physiques, les protocoles utilisés, les communications qui ont lieu. La simulation doit d'abord être saisie sous forme de fichier que NS va utiliser pour produire un fichier contenant les résultats. Mais l'utilisation de l'OTCL permet aussi à l'utilisateur de créer ses propres procédures (par exemple s'il souhaite enregistrer dans un fichier l'évolution d'une variable caractéristique du réseau au cours du temps). Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage unicast ou multicast, des protocoles de transport, de session, de réservation, des services intégrés, des protocoles d'application comme FTP. A titre d'exemple la liste des principaux composants actuellement disponibles dans NS par catégorie est :

•application : Web, ftp, Telnet, générateur de trafic (CBR...).

•transport : TCP, UDP, RTP, SRM.

•routage unicast : Statique, dynamique (vecteur distance).

•routage multicast : DVMRP, PIM.

•gestion de file d'attente : RED, DropTail, Token bucket..

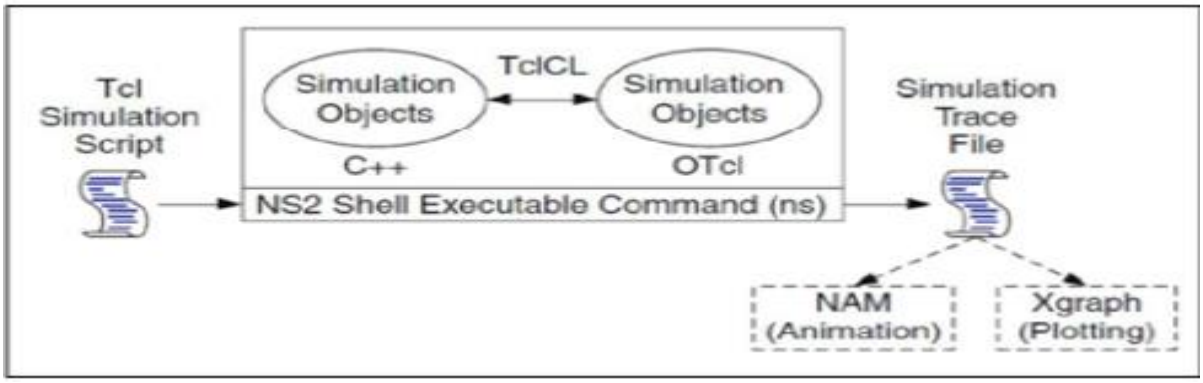


Figure 27 : L'architecture du simulateur NS2.[2]

4.4. L'outil de visualisation

NS-2 ne permet pas de visualiser le résultat des expérimentations. Il permet uniquement de stocker une trace de la simulation, de sorte qu'elle puisse être exploitée par un autre logiciel, comme NAM.

NAM est un outil de visualisation qui présente deux intérêts principaux : représenter la topologie d'un réseau décrit avec NS-2, et afficher temporellement les résultats d'une trace d'exécution NS-2. Par exemple, il est capable de représenter des paquets TCP ou UDP, la rupture d'un lien entre noeuds, ou encore de représenter les paquets rejetés d'une file d'attente pleine. Ce logiciel est souvent appelé directement depuis les scripts TCL pour NS-2, pour visualiser directement le résultat de la simulation.

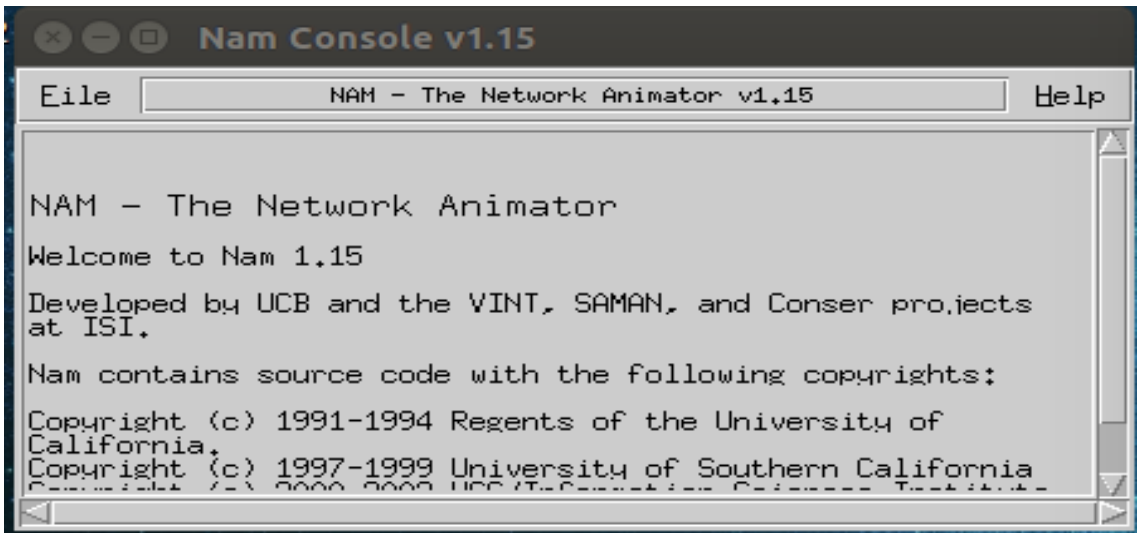


Figure 28 : la console NAM.

4.5. Paramètres de simulation

Nous avons besoin pour simuler le protocole de routage d'un ensemble de paramètres réseau (qui sont très nombreux), les valeurs des paramètres de nos simulations qui ne seront pas varié sont présentées dans la table suivante :

Paramètres	Valeur
Couche MAC	IEEE 802.11
Type d'antenne radio	Omni Antenna
Débit de communication	2 MB /s
Portée de transmission	250 mètres
Nombre maximum de noeuds	50
Nombre de trafics utilisés	25
Surface de simulation	1000x500 m2
Durée de la simulation	300 secondes
Modèle de mobilité	Random Waypoint
Taille des paquets	50 octets
Taux de transmission	4 parquets/s
Duré de pause pour une variation de vitesse	[50,150] s
Modèle de communication	CBR sur UDP

Table 6 : **Paramètres de simulations**.

4.6. Avantages et inconvénients de la simulation

✓ **Avantage :**

•Observations des états du système.

•Etudes des points de fonctionnement d'un système.

•Etudes de l'impact des variables sur les performances du système.

•Etude d'un système sans les contraintes matérielle.

✓ **Inconvénients :**

•La conception de modèles peut nécessiter des compétences spéciales.

•Résultats pas forcément généralisable.

4.7. Installation du simulateur NS2

- **Étape 1 :**

  Tout d'abord, téléchargez Network Simulator (NS-2.35). Après, nous devons maintenant mettre à jour Ubuntu avec ses derniers composants. Ouvrez un terminal et exécutez ces commandes :

- **Sudo apt-get install tcl8.5-dev tk8.5-dev**
- **Sudo apt-get install build-essential autoconf automake**
- **Sudo apt-get install perl xgraph libxt-dev libx11-dev libxmu-dev**

- **Étape 2 :**

Téléchargez-le package NS2 à partir de ce lien
https://sourceforge.net/projects/nsnam/postdownload.
Copiez le fichier téléchargé dans votre dossier Home dans Ubuntu 14.04. Faites un clic droit sur le fichier et sélectionnez l'option "Extraire ici".

- **Étape 3 :**

Allez maintenant dans le sous-dossier ns-allinone-2.35 / ns-2.35 / linkstate et faite double-clique sur le fichier "ls.h" pour l'ouvrir et allez à la ligne 137 et changez la ligne ci-dessous.

- **from**
- **void eraseAll () { erase (baseMap::begin(), baseMap::end()); }**
-
- **to**
- **void eraseAll() { this->erase(baseMap::begin(), baseMap::end()); }**

- **Étape 4 :**

Ouvrez le terminal en appuyant sur la combinaison de touches "ALT + CNTL + T". Et passez au dossier ns-allinne-2.35 via le terminal.

marwa@marwa -PC:~/ns-allinone-2.35$ ./install

Appuyez sur Entrée et attendez un certain temps jusqu'à ce qu'il affiche les informations de chemin. C'est fait maintenant et vous êtes installé NS2.

- **Étape 5 :**

Il est maintenant temps de définir les informations de chemin. Dans le terminal, utilisez sudo gedit. bashrc et appuyez sur Entrée. Il demandera un mot de passe pour entrer (ce n'est pas visible).

marwa@marwa -PC:~$ sudo gedit .bashrc
[sudo] password for marwa:

Allez à la dernière ligne du fichier nouvellement ouvert (bashrc), copiez et collez ces 3 lignes. Assurez-vous que vous avez changé marwa avec votre nom d'utilisateur sur Ubuntu.

PATH=$PATH:/home/ marwa /ns-allinone-2.35/bin:/home/ marwa /ns-allinone-2.35/tcl8.5.10/unix:/home/ marwa /ns-allinone-2.35/tk8.5.10/unix

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/ marwa /ns-allinone-2.35/otcl-1.14:/home/ marwa /ns-allinone-2.35/lib

TCL_LIBRARY=$TCL_LIBRARY:/home/ marwa /ns-allinone-2.35/tcl8.5.10/library

Enregistrez le document et fermez. Rechargez-le. Bashrc à l'aide de la commande suivante.

source ~/.bashrc

- **Étape 6 :**

C'est fait ! ouvrez le terminal et tapez "ns" appuyez sur Entrée. Vous obtiendrez un signe%, il indique l'installation est réussie.

4.8. Installation du Tracegraph 2.02

Tracegraph est une excellente application qui est pratique pour les utilisateurs de NS2. Il élimine le besoin de configurer et d'exécuter des scripts perl / awk sur le fichier de trace. Analyse des fichiers de trace simplifiée. Bien que je pense que Tracegraph en est encore à ses balbutiements, sa portée actuelle fournit juste tout ce dont un chercheur utilisant NS2a besoin. Tout d'abord, pour commencer l'installation il faut téléchargez les packages suivants :

www.mediafire.com/file/z0o0ma44ovb8ohj/tracegraph202linux.tar.gz

www.mediafire.com/file/c9735hyvbnrdc1e/mglinstaller.gz

Tracegraph semble avoir été développé en utilisant Matlab et donc un code de support est nécessaire pour le faire fonctionner sous Ubuntu. C'est la raison pour laquelle l'installation de mglinstaller.

- Extrayez tracegraph202linux.tar.gz dans votre dossier personnel. Dans mon cas, cela donnerait / home / marwa / tracegraph202.
- Extrayez ensuite mglinstaller.gz dans / home / marwa / tracegraph202, Un seul exécutable nommé mglinstaller apparaîtrait dans le dossier tracegraph202.
- Ensuite, fournissez l'autorisation exécutable à mglinstaller et exécutez-le à l'aide de la commande suivante :

$ sudo export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/marwa/tracegraph202/bin/glnx86

Maintenant, accédez à cet emplacement :

$ cd /home/ marwa /tracegraph202/bin/glnx86

Vous verriez un exécutable nommé trgraph à cet emplacement. Fournissez l'autorisation exécutable à ce fichier et exécutez-le.

- **$ sudo su**
- **# chmod 777 trgraph**
- **# ./trgraph**

Pour exécuter tracegraph à chaque fois, accédez simplement à / home / marwa / tracegraph202 / bin / glnx86 dans Terminal et exécutez trgraph comme suit :

- **$ ./trgraph**

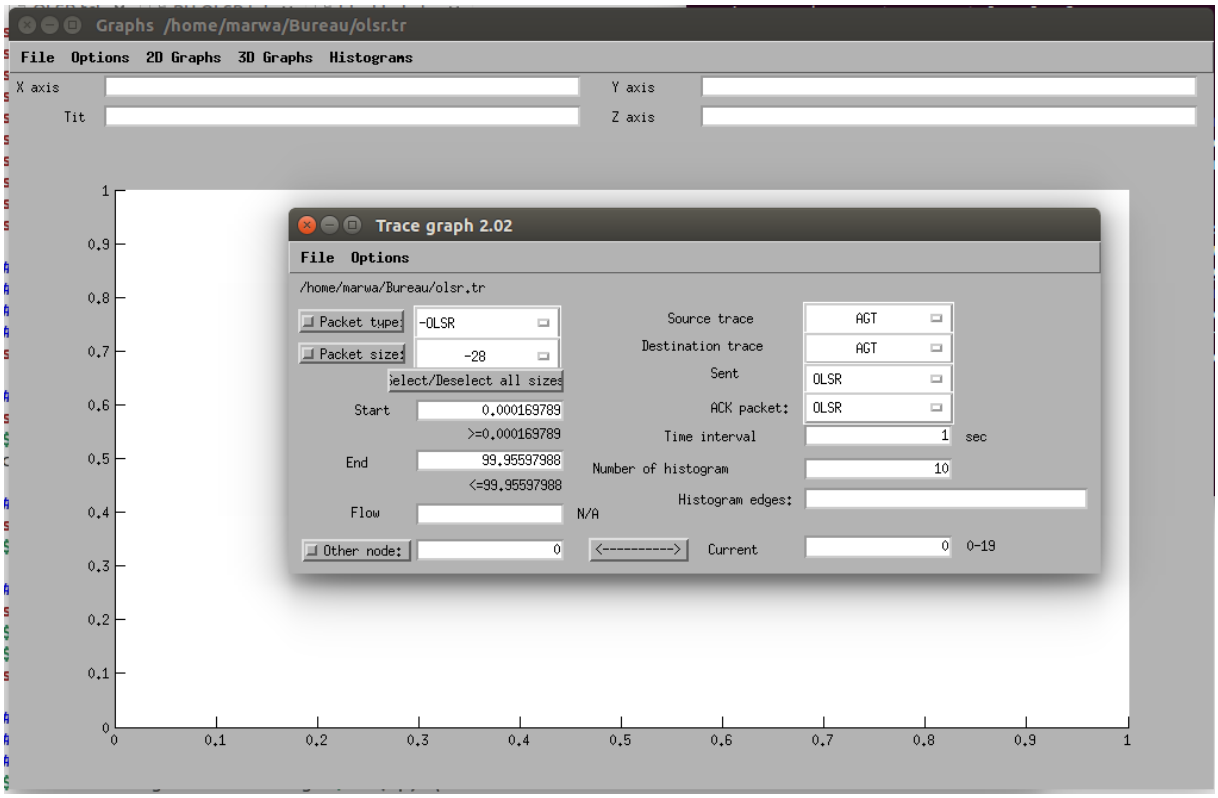


Figure 29 : Format du tracegraph 2.02.

4.9. Environnement de développement

Notre système est développé sous l'environnement :

- **Système d'exploitation: Ubuntu**14.04
- **Environnement de programme:** NS2.35
- **Interface graphique:** Xgraph
- **Ordinateur portable: Intel** (R) Core(TM) i5-3210M CPU @ 2.50 GHz 2.50 GHz , RAM: 8Go

4.10. Variables de simulation

Nous étudions les performances du protocole **OLSR et BH-OLSR** avec l'attaque BLACKHOLE et notre solution proposé SU-OLSR par rapport au temps. Nos variables de simulation sont **:**

- Les paquets envoyés
- Les paquets reçus
- Les paquets perdus.

4.11. Analyse des attaques BH-OLSR et W-OLSR contre OLSR

4. 11.1. Simulation du protocole OLSR :

Pour étudier la simulation du protocole de routage OLSR et BH-OLSR sur le réseau nous avons pris en considération la vitesse de déplacement varie entre [1, 100] m/s et nombre des nœuds fixé à (20 nœuds).

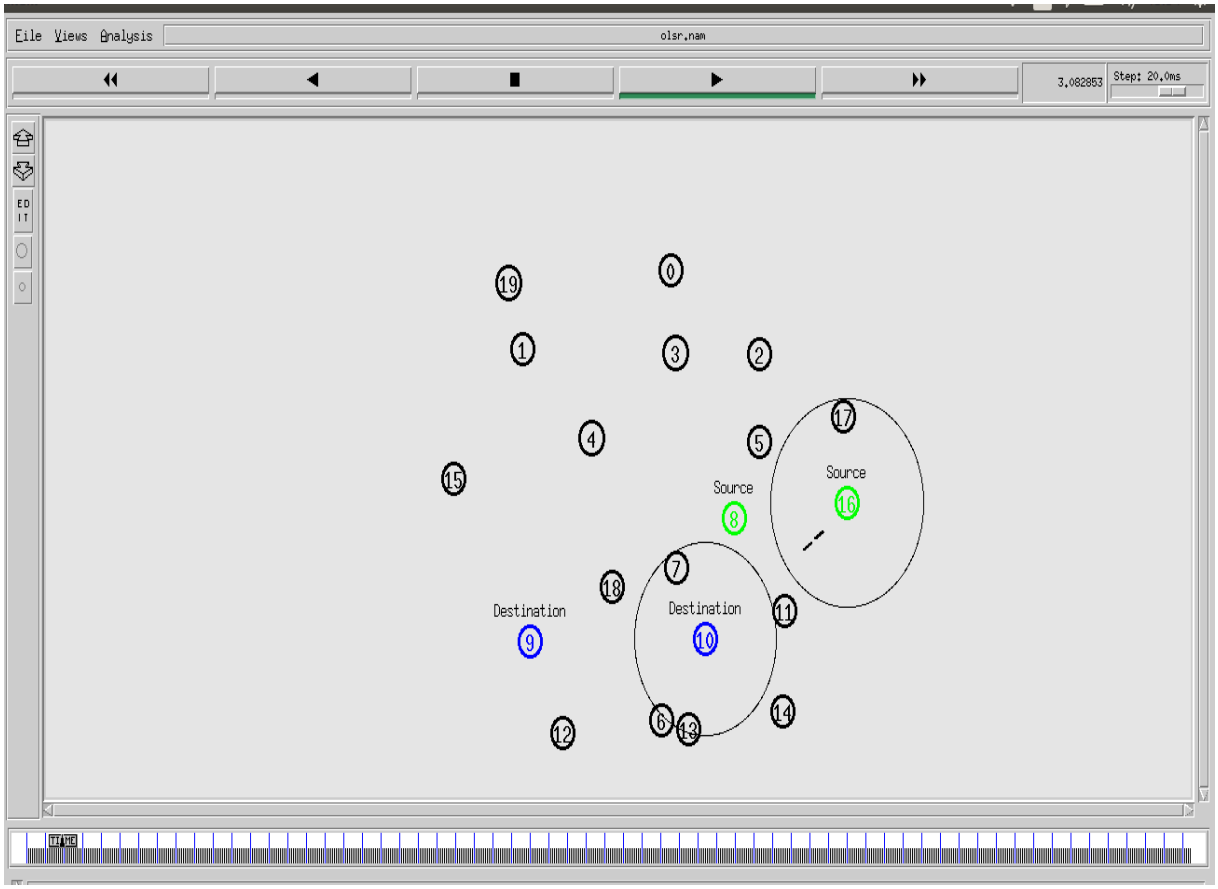


Figure 30 : Simulation du protocole OLSR.

4. 11.2. Simulation du protocole BH-OLSR :

Pour étudier la simulation du protocole BH-OLSR sur le réseau nous avons intégré l'attaque BLACKHOLE sur notre simulation du protocole OLSR pour aboutir à une comparaison entre ces deux derniers.

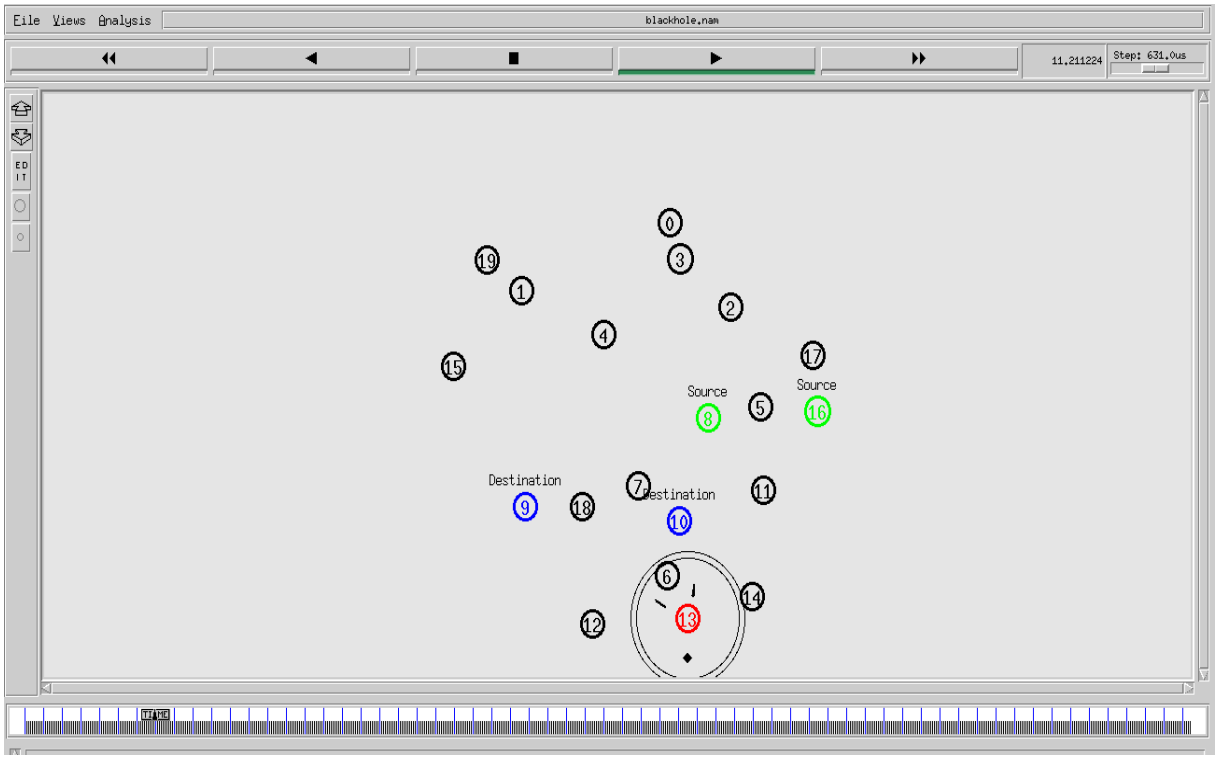


Figure 31 : la simulation du protocole BH-OLSR avec tracegraph.

L'attaque BLACKHOLE a deux propriétés :

Tout d'abord, le nœud exploite le protocole de routage ad hoc, tel que le protocole OLSR, pour s'annoncer comme ayant une route valide vers un nœud de destination. Le nœud qui exploite la route du protocole OLSR est faux, qui a l'intention d'intercepter des paquets.

Deuxièmement, le nœud qui exploite la route du protocole OLSR consomme les paquets interceptés.

➢ Les paquets envoyés :

OLSR **BH-OLSR**

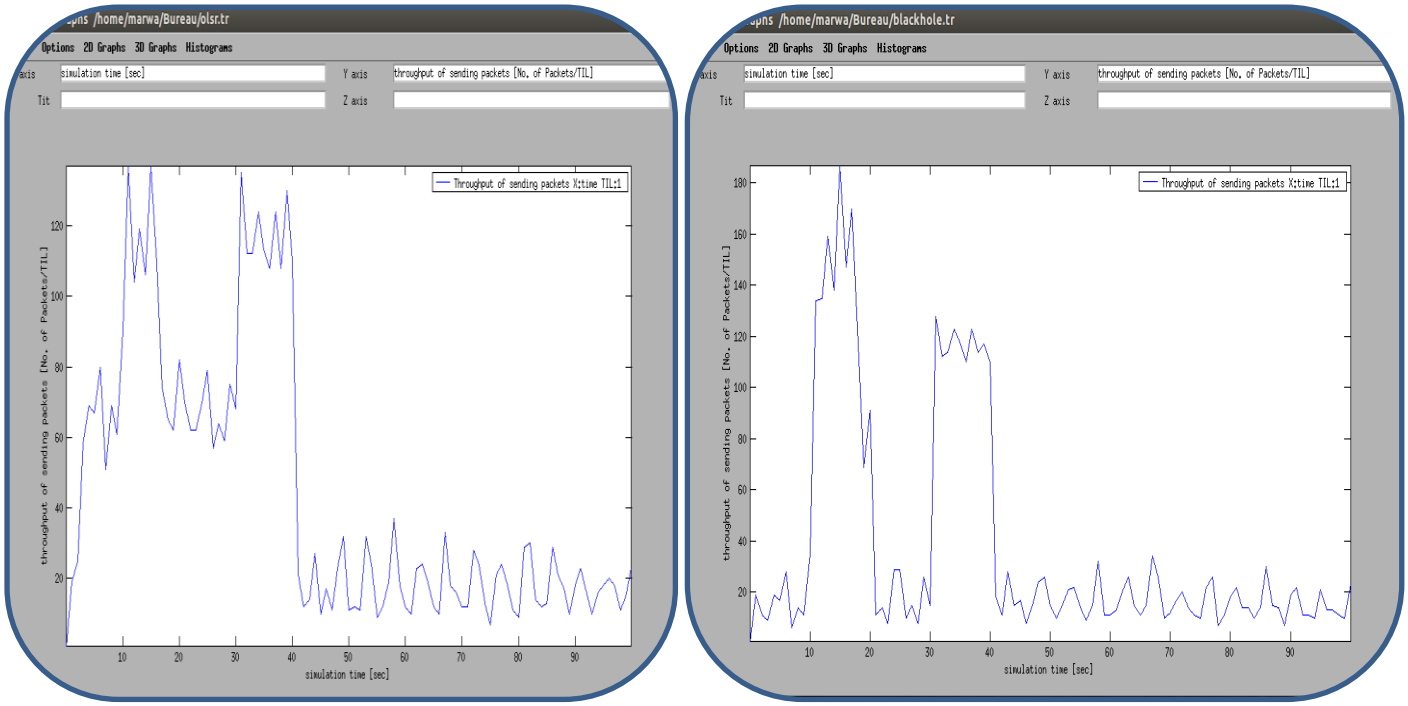


Figure 32 : Les protocoles OLSR et BH-OLSR selon les paquets envoyés.

La figure 33 montre une évolution décroissante du nombre des paquets envoyés en fonction du temps. En effet, l'attaque BLACKHOLE provoque un changement de la topologie qui doit être géré par les noeuds en recalculant les tables de routage. A travers le graphe, nous pouvons voir que le nombre des paquets envoyés des protocoles **OLSR** est nettement supérieur à celui du protocole de base **BH-OLSR**.

➢ Les paquets reçus :

OLSR **BH-OLSR**

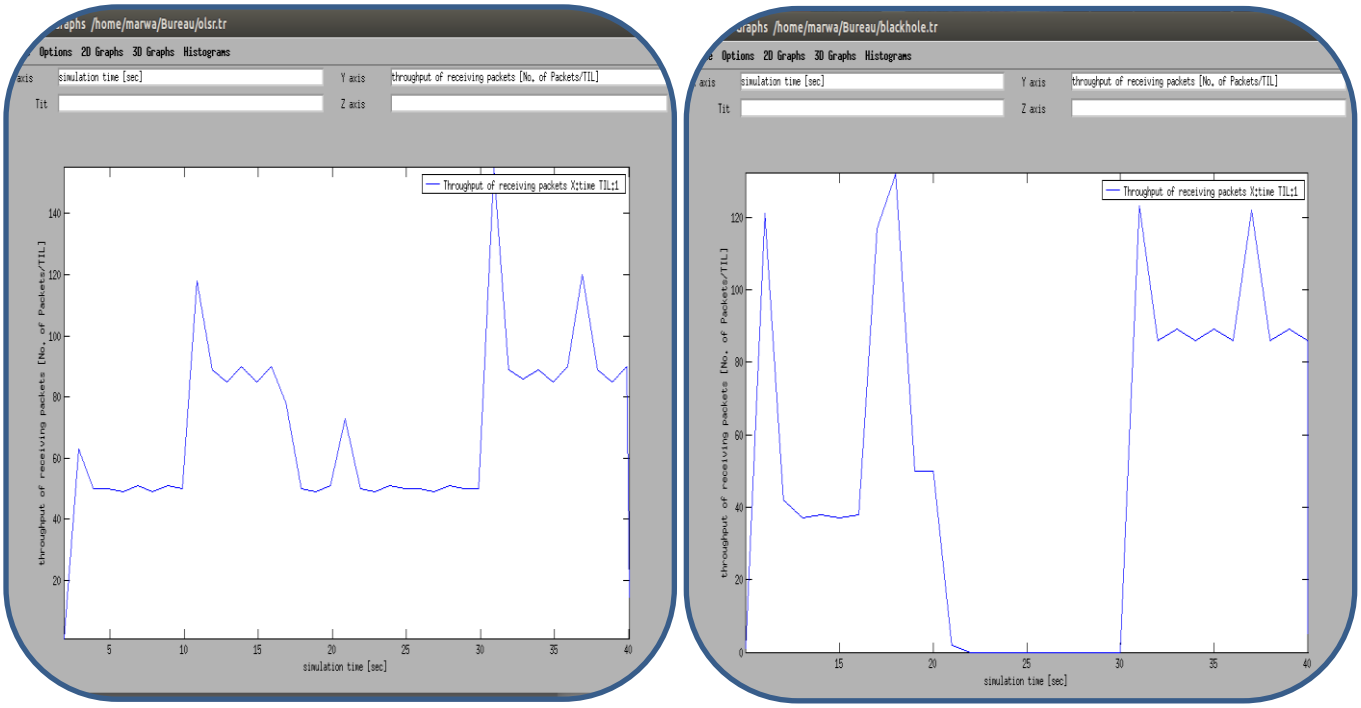


Figure 33 : Les protocoles OLSR et BH-OLSR selon les paquets reçus.

Dans cette figure 34 montre une diminuation du nombre des paquets reçus en fonction du temps. En effet, l'attaque BLACKHOLE a stoppé la réception des messages dans l'intervalle du 20s à 30s, donc les MPR ont été touché par l'attaque, par conséquent les message TC aussi. A travers le graphe, nous pouvons voir que le nombre des paquets reçus des protocoles **OLSR** est nettement supérieur à celui du protocole de base **BH-OLSR**.

➢ Les paquets perdus :

OLSR **BH-OLSR**

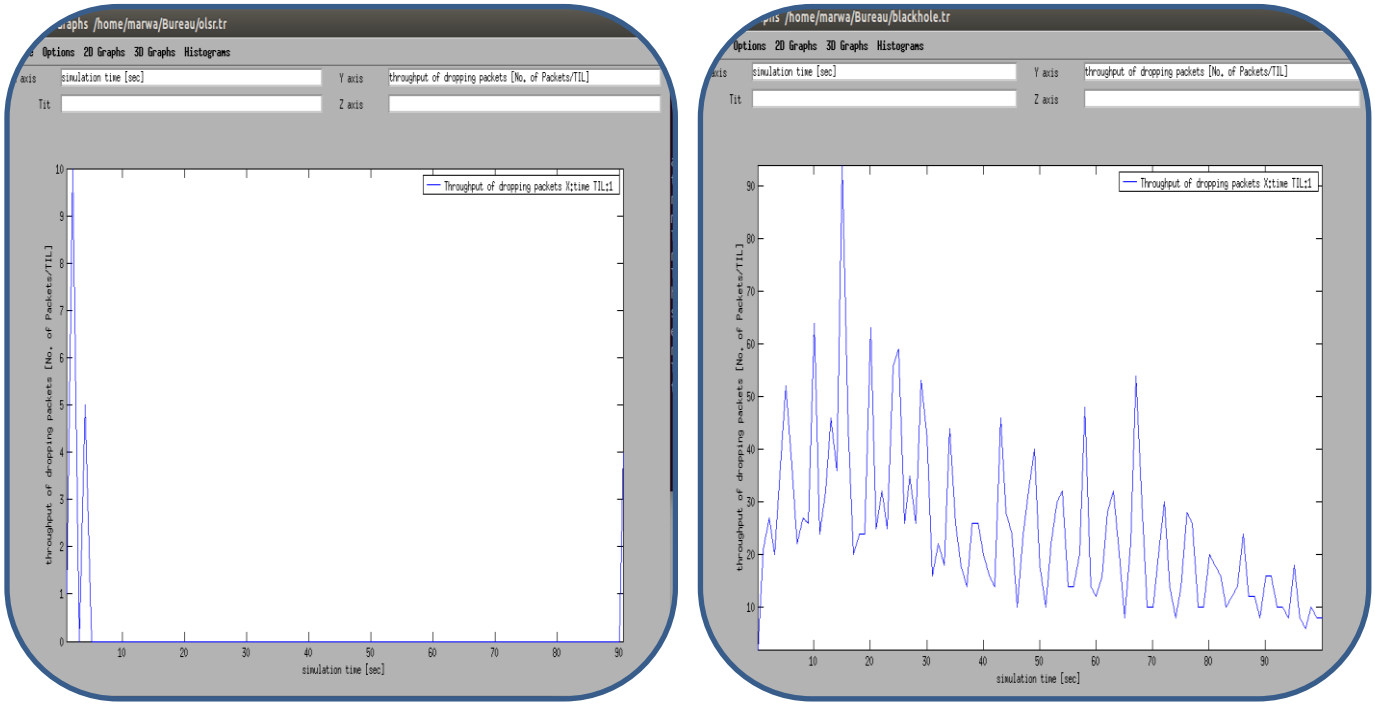


Figure 34 : Les protocoles OLSR et BH-OLSR selon les paquets perdus.

D'après la figure ci-contre, nous remarquons une grande évolution de la perte des paquets perdus du protocole **BH-OLSR** par rapport du protocole **OLSR** dès le début de la simulation jusqu'au la fin à cause de l'intégration de l'attaque BLACKHOLE qui a détruit les paquets **OLSR**. Donc le mécanisme de sécurité qui doit être ajouté dans **BH-OLSR** doit influencer le mécanisme d'établissement des routes.

OLSR **BH-OLSR**

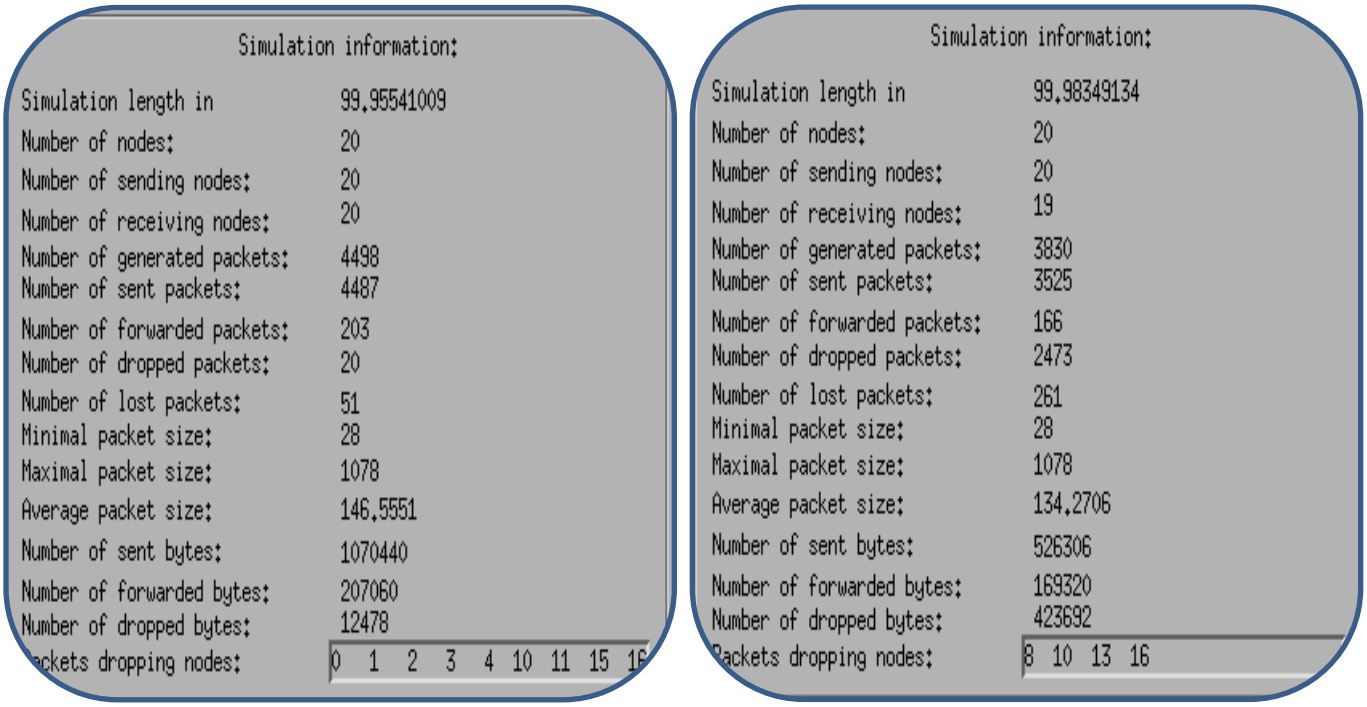


Figure 35 : Les informations de la simulation du OLSR et BH-OLSR.

4. 11.3. Simulation du protocole W-OLSR :

Pour étudier la simulation du protocole W-OLSR sur le réseau nous avons intégré l'attaque WORMHOLE sur notre simulation du protocole OLSR pour aboutir à une comparaison entre ces deux derniers.

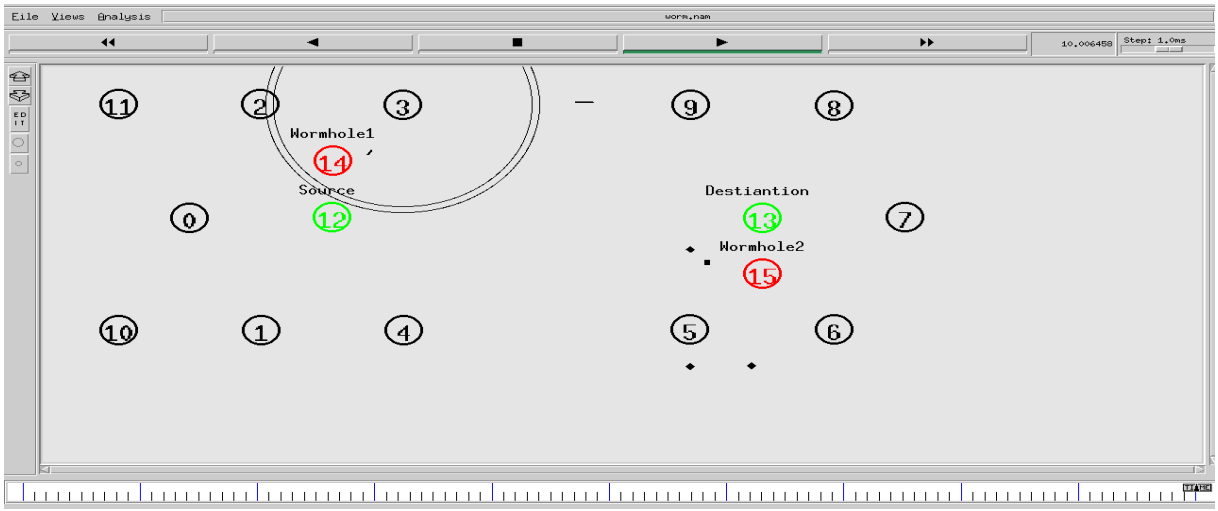


Figure 36 : la simulation du protocole W-OLSR avec tracegraph.

➢ Les paquets envoyés :

OLSR **W-OLSR**

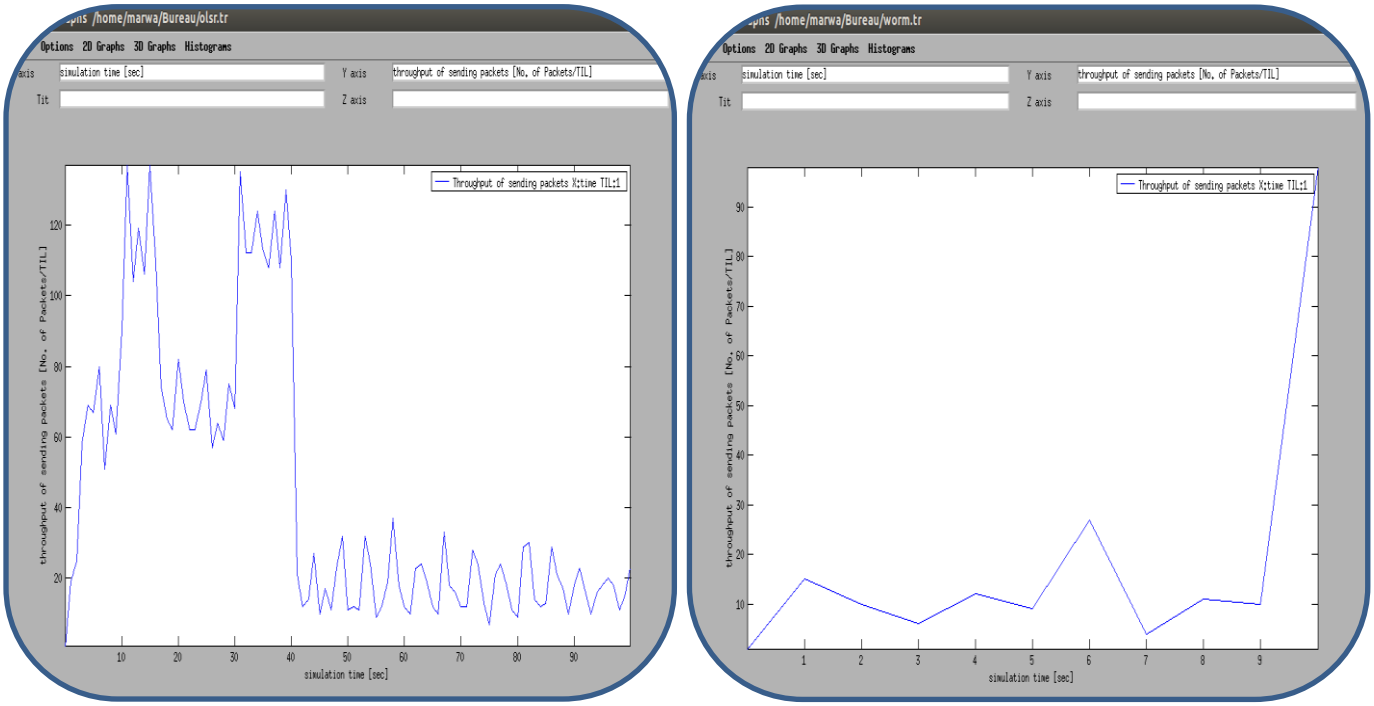


Figure 37 : Les protocoles OLSR et W-OLSR selon les paquets envoyés.

D'après le graphe de la figure 38, nous remarquons une grande Diminution significative de l'envoi des paquets du protocole **W-OLSR** par rapport du protocole **OLSR** dès le début de la simulation jusqu'au la fin à cause de l'intégration de l'attaque WORMHOLE qui a détruit les paquets **OLSR** Et elle est classé comme l'une des attaques les plus dangereuses.

➢ Les paquets reçus :

OLSR **W-OLSR**

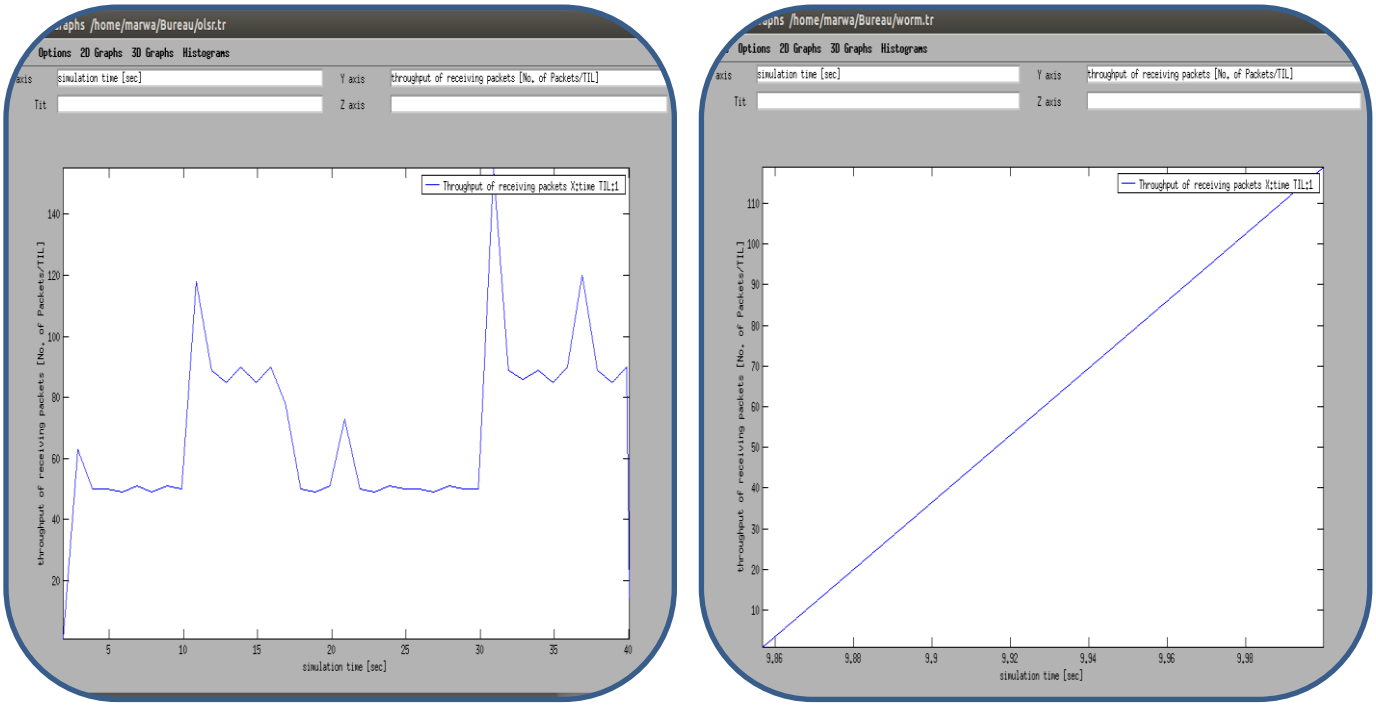


Figure 38 : Les protocoles OLSR et W-OLSR selon les paquets reçus.

D'après les courbes des graphes de la figure 39, Le protocole OLSR présente un taux de réception de paquets plus élevé par rapport à la courbe du protocole W- OLSR. En effet, Ceci s'explique par le fait que l'attaque est active, et les messages de l'attaquant sont transmis par les noeuds du réseau qui découvrent et on constate que le protocole **W-OLSR** recoit les paquets que dans l'intervalle du 9,86s à 9,98s.

➢ Les paquets perdus :

OLSR **W-OLSR**

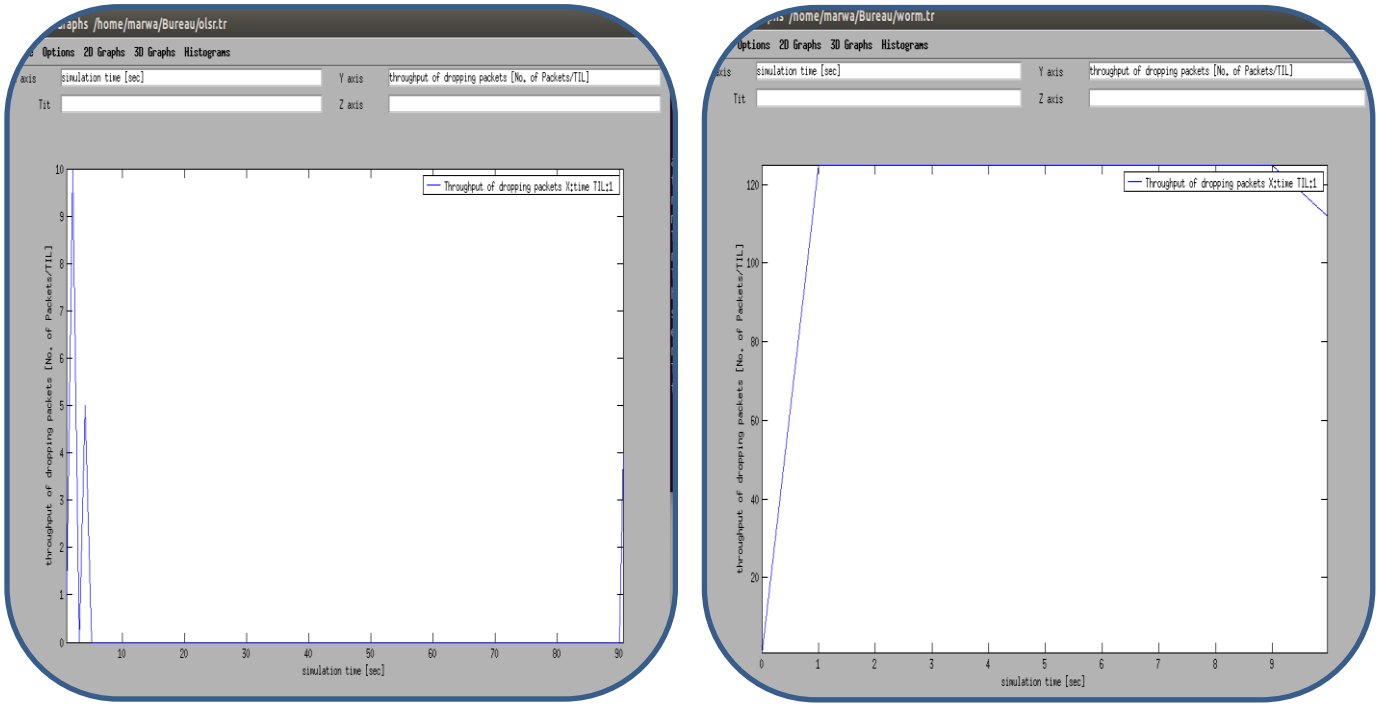


Figure 39 : Les protocoles OLSR et W-OLSR selon les paquets perdus.

D'après la figure 40, nous pouvons faire les mêmes constatations que celles faites dans les simulations précédentes en présence de l'attaque. Nous remarquons une grande évolution de la perte des paquets perdus du protocole **W-OLSR** par rapport du protocole **OLSR** dès le début de la simulation jusqu'au la fin à cause de l'intégration de l'attaque WORMHOLE qui a détruit les paquets **OLSR**. Donc le mécanisme de sécurité qui doit être ajouté dans **W-OLSR** doit influencer le mécanisme d'établissement des routes.

4.12. Analyse de l'attaque BLACKHOLE et la solution SU-OLSR

➢ Les paquets envoyés :

BH-OLSR **SU-OLSR**

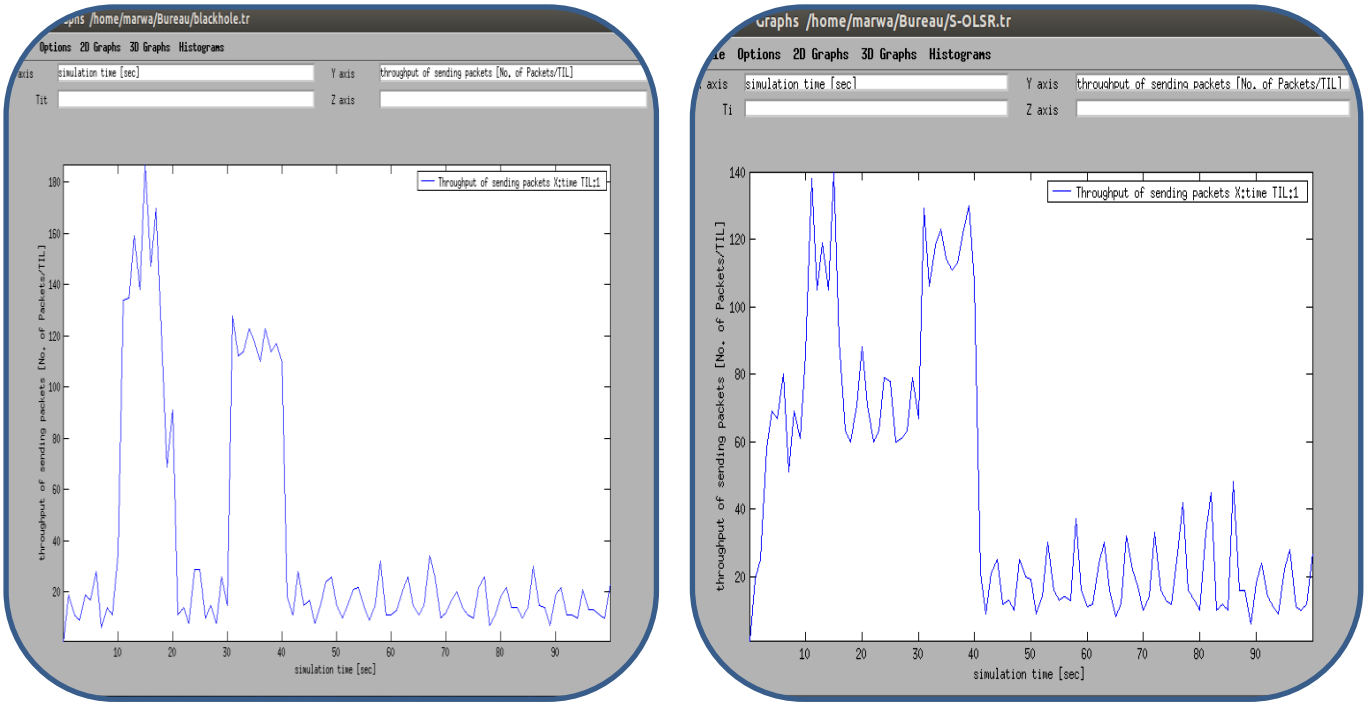


Figure 40 : Les protocoles BH-OLSR et SU-OLSR selon les paquets envoyés.

Nous avons intégré la solution de sécurité SU-OLSR pour protéger le protocole étudié contre l'attaque BLACKHOLE. La figure **41** montre le comportement du réseau en termes de nombre de paquets envoyés les deux cas de figures simulées **BH-OLSR et SU-OLSR**. Nous remarquons un comportement normal du nombre de paquets envoyés du protocole **SU-OLSR**, car il augmente avec le temps et, la valeur affichée par **SU-OLSR** dépasse de peu celle affichée par **OLSR** ou **BH-OLSR**. Donc nous constatons qu'on a presque obtenu un état comme l'état initiale du protocole OLSR et c'est notre but d'étude.

➢ Les paquets reçus :

BH-OLSR **SU-OLSR**

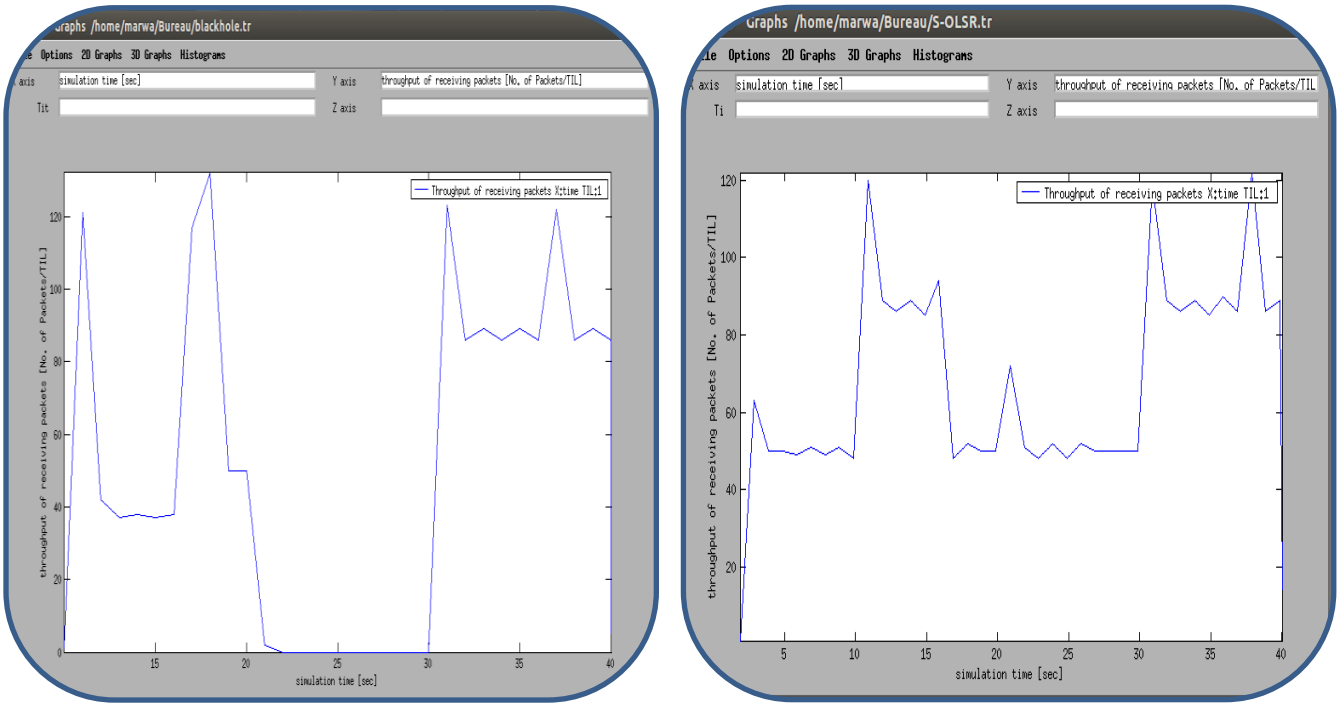


Figure 41 : Les protocoles BH-OLSR et SU-OLSR selon les paquets reçus.

La figure 42 montre le nombre de paquets reçus durant la simulation. Pour le cas de figures **SU-OLSR**, ce paramètre augmente avec le temps. Mais, la valeur affichée par **SU-OLSR** dépasse de peu celle affichée par **OLSR** ou **BH-OLSR**. Ce comportement est justifié par le fait que **SU-OLSR** élimine volontairement plus de paquets que OLSR ou **BH-OLSR.**

➢ Les paquets perdus :

BH-OLSR **SU-OLSR**

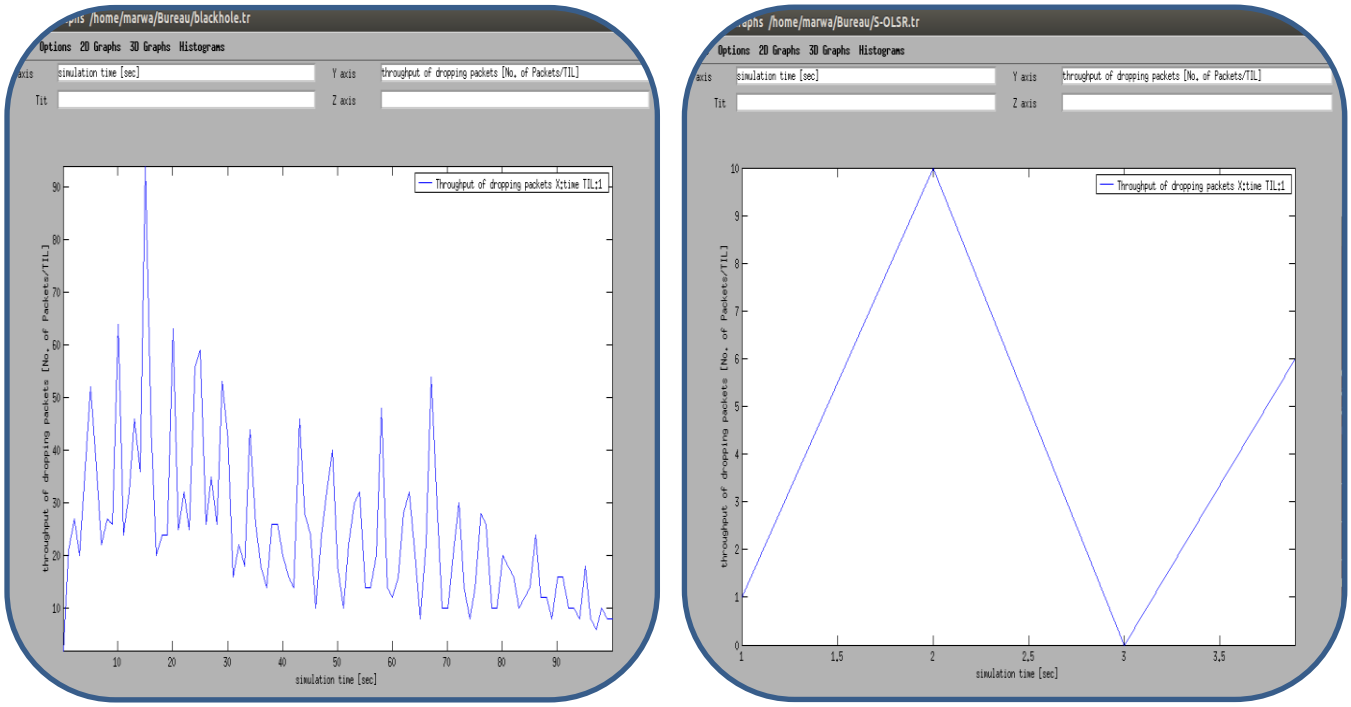


Figure 42 : Les protocoles BH-OLSR et SU-OLSR selon les paquets perdus.

En analysant les deux graphes de la figure 43, l'allure des courbes reflète bien l'amélioration apportée par notre approche en termes de réduction du nombre de paquets perdus par rapport au protocole **BH-OLSR**. Le protocole **SU-OLSR** enregistre un taux de perte faible par rapport à **BH-OLSR** et notre protocole OLSR, ce qui est évident, vu que ce dernier n'introduit pas des mécanismes de sécurité.

BH-OLSR

Simulation information:

Simulation length in	99.98349134
Number of nodes:	20
Number of sending nodes:	20
Number of receiving nodes:	19
Number of generated packets:	3830
Number of sent packets:	3525
Number of forwarded packets:	166
Number of dropped packets:	2473
Number of lost packets:	261
Minimal packet size:	28
Maximal packet size:	1078
Average packet size:	134.2706
Number of sent bytes:	526306
Number of forwarded bytes:	169320
Number of dropped bytes:	423692
Packets dropping nodes:	8 10 13 16

SU-OLSR

Simulation information:

Simulation length in	99.98091342
Number of nodes:	20
Number of sending nodes:	20
Number of receiving nodes:	19
Number of generated packets:	4529
Number of sent packets:	4518
Number of forwarded packets:	195
Number of dropped packets:	17
Number of lost packets:	72
Minimal packet size:	28
Maximal packet size:	1078
Average packet size:	145.6828
Number of sent bytes:	1069550
Number of forwarded bytes:	198900
Number of dropped bytes:	12032
Packets dropping nodes:	0 1 3 4 7 15 16

Figure 43 : Les informations de la simulation du BH-OLSR et SU-OLSR.

La figure 44 montre une évolution croissante de nos variables de simulation ; nombre de paquets envoyés ; nombre de paquets reçu et nombre de paquets perdus, En effet, la mobilité des noeuds provoque un changement de la topologie qui doit être géré par les noeuds en recalculant les tables de routage. A travers le graphe ; nous pouvons voir que nos variables de simulation des protocoles **SU-OLSR** est nettement supérieur à celui du protocole de base **BH-OLSR et OLSR**. Cela est dû au temps requis par les procédures cryptographiques (génération et vérification des signatures), ainsi que le calcul et la vérification des éléments de hachage pour le protocole, qui rend la procédure d'établissement des routes plus lente. Nous constatons aussi que la courbe de **SU-OLSR** permet d'accélérer le processus de calcul des routes, et par conséquent, l'acheminement des paquets à leurs destinations se fait plus rapidement. Nous concluons donc que notre protocole **SU-OLSR** établit ses routes plus rapidement que le protocole **BH-OLSR et OLSR**, ce qui le rend plus performant en termes de performances de routage.

4.13. Conclusion

Dans cette partie nous avons évalué les performances de notre protocole SU_OLSR en le comparant aux deux autres protocoles BH-OLSR et OLSR. A travers les courbes obtenues, nous remarquons bien l'influence de la variation de temps sur les métriques évaluées. Les simulations ont montré que SU_OLSR est plus performant en termes de performances de routage et de coût que le protocole OLSR. En effet, le mécanisme des MPR de confiance et paquet OLSR de confiance permet d'offrir plus d'efficacité et plus de rapidité à la solution et diminue considérablement le coût de sécurité.

Nous pouvons conclure que le protocole SU_OLSR se comporte globalement d'une manière assez satisfaisante par rapport au protocole OLSR. Il offre un taux de perte de paquets plus faible, accélère les fonctionnalités de routage : le calcul des routes et l'acheminement des paquets se font plus rapidement que le protocole OLSR. Cela contribue énormément à l'amélioration des performances du réseau tout en assurant un bon niveau de sécurité à moindre coût.

Nous avons étudié également les performances du modèle proposé à travers la simulation de son comportement en présence de deux types d'attaques : l'attaque BLACKHOLE ainsi que l'attaque WORMHOLE. Les attaques dégradent les performances des protocoles de routage, et affectent significativement le bon fonctionnement du réseau. Les courbes du protocole OLSR obtenues en présence d'attaques montrent clairement cette limite. Par contre, ces différentes simulations ont montré que le protocole SU_OLSR a résisté aux attaques, et il a réussi à détecter les messages fabriqués et rejoués limitant ainsi l'impact de ces attaques sur les performances du réseau. D'où la nécessité d'élaborer des extensions de protocoles de routage intégrant des mécanismes de sécurité robustes afin de protéger le réseau et faire face efficacement aux attaques.

La solution proposée semble offrir une possibilité réelle pour contrer les attaques sans dégrader les performances du routage de façon très prononcée. Cependant l'étude de notre contribution a besoin d'être élargie avec des simulations sur d'autres paramètres afin de l'évaluer de façon plus précise et plus approfondie.

CONCLUSION GENERALE

La sécurisation du routage dans les réseaux Ad hoc reste un problème majeur. Elle se heurte souvent à la difficulté de proposer des mécanismes relativement robustes face aux différentes attaques possibles, causées par les intrusions externes et les nœuds compromis sans pour autant affecter les performances globales du réseau ad hoc et des protocoles de routage de manière trop prononcée.

Dans ce travail, nous avons étudié les problèmes de sécurité dans les protocoles de routage des réseaux mobiles Ad hoc d'un point de vue théorique. Cette étude a révélé un nombre de difficultés liées à l'absence d'infrastructure centralisée, la contrainte d'énergie, la topologie dynamique, la bande passante, les ressources limitées, etc. De nombreux travaux de recherche proposent des schémas de sécurité qui conviennent aux caractéristiques des réseaux ad hoc. Bien que des solutions répondent à un ensemble d'exigences de sécurité, il n'en demeure pas moins que les solutions les plus efficaces et les plus complètes sont coûteuses.

Notre travail focalisait sur la sécurisation du protocole de routage OLSR, dans le but de proposer une solution de sécurité efficace qui permette de préserver les performances globales du réseau. Nous étions attirés par l'efficacité et la rapidité de traitement de notre solution. En effet, cette approche est beaucoup plus légère et beaucoup moins coûteuse que la signature numérique. L'idée de base de notre solution était de mettre en place un système de confiance à base de vérification des nœuds suspects qui permet aux noeuds voisins d'échanger les informations de routage de façons sécurisée. Pour plus d'efficacité, nous avons intégré notre solution dans une extension de sécurité du protocole OLSR déjà existante. Nous avons choisi le protocole BH-OLSR qui est un protocole OLSR en introduisant l'attaque BLACKHOLE qui souffre de surcoûts très élevés induits par le grand nombre de paquets perdus. Ces surcoûts entraînent une forte dégradation des performances de BH-OLSR et sont à l'origine de pertes notables de paquets.

L'intégration de notre schéma de sécurité dans le protocole BH-OLSR a permis aux noeuds voisins d'échanger les messages de contrôle HELLO de manière sécurisée, efficace, et moins coûteuse. En effet, notre mécanisme de sécurité a permis de réduire le nombre d'opérations cryptographiques utilisées pour le calcul de la signature numérique, diminuant ainsi les surcoûts

de calcul associés au traitement des messages. Cela a permis de réduire le nombre de paquets perdus, et de préserver les ressources des noeuds, prolongeant ainsi, la durée de vie du réseau. Un autre point fort de notre solution est que la facilité et la rapidité de la manipulation des nœuds malicieux permettent un gain d'espace et de temps non négligeable, ce qui permet d'accélérer les fonctionnalités de routage, notamment, la génération et le traitement des messages HELLO, la mise à jour des différentes tables topologiques et le calcul des routes. Toutes ces améliorations permettent d'optimiser considérablement les performances du protocole BH-OLSR, et par conséquent, cela contribuera énormément à l'amélioration des performances du réseau et son bon fonctionnement. L'objectif de notre approche est aussi d'empêcher d'une part les attaques sur les messages de contrôle, et de fournir d'autre part, un support fiable pour la détection des comportements malveillants. Pour cela, nous avons développé une deuxième ligne de défense qui consiste en un mécanisme de sécurité à la fois préventif et réactif pour contrer l'attaque du trou de ver. Une attaque qui peut être initiée par un noeud externe malveillant pour ensuite mener d'autres attaques plus dangereuses, à savoir l'attaque du trou noir dont les conséquences sont très lourdes et désastreuses sur le fonctionnement du réseau. Cette attaque entraîne des erreurs dans le processus de calcul des routes, et conduit à l'établissement de routes corrompues et incorrectes, une perte de connectivité et dégradation de la communication entre les noeuds du réseau.

Le mécanisme que nous avons proposé permet d'augmenter le niveau de sécurité et de renforcer la robustesse de notre solution face à ce type d'attaques. En effet, ce mécanisme permettra de détecter l'occurrence de ces deux attaques dans le réseau, et d'isoler les noeuds malveillants qui en sont responsables, afin d'empêcher leur participation dans les opérations de routage et renforcer la construction de routes plus fiables et plus sûres. Les résultats de la simulation montrent clairement que le schéma de sécurité que nous avons proposé réalise un compromis entre la robustesse et l'efficacité en termes de sécurité et les performances globales du réseau. Notre approche est donc bien adaptée au contexte ad hoc. Elle présente une bonne solution pour sécuriser l'échange des informations de routage tout en respectant les contraintes et les limitations imposées par cet environnement. Les travaux réalisés dans ce mémoire nous ont permis de tracer les perspectives suivantes :

Nous envisageons de valider par simulation le deuxième mécanisme de sécurité que nous avons proposé. Afin d'étudier de façon expérimentale ces performances et sa robustesse en présences d'attaques.

Afin d'améliorer les performances de sécurité de notre approche, nous devrons l'enrichir par des mécanismes de sécurité plus avancés. Une solution intéressante serait la combinaison avec

un système de réputation plus sophistiqué qui va permettre d'augmenter le niveau de protection pour conter efficacement des attaques plus complexes. A savoir, des attaques où plusieurs adversaires coopèrent pour tenter de déséquilibrer le routage, ou bien, les menaces pouvant venir des noeuds internes légitimes qui refusent de coopérer dans les opérations de routage dans un souci de préservation de leurs ressources énergétiques. Or de tels comportements portent atteinte au fonctionnement global du réseau. Car ils peuvent provoquer la formation de chemins incorrects, des pertes de connectivité entre les entités, ou une moins bonne répartition du trafic (réduisant potentiellement la durée de vie du réseau).

Dans ce système de réputation tous les noeuds du réseau coopèrent entre eux en échangeant leurs valeurs de réputations. Ainsi, l'évaluation du comportement de chaque noeud ne se basera pas seulement sur les données recueillies localement par chaque noeud, mais elle impliquera également l'information de réputation fournie par les autres noeuds du réseau. Cette collaboration entre les noeuds aura pour effet l'augmentation du niveau de précision et la fiabilité de l'évaluation. Cela permet de donner une certaine confiance aux noeuds qui ont un bon comportement et d'éliminer les noeuds dont l'action malveillante a été observée.

Un autre point fort de la mise en place d'un système de réputation, est que les fonctionnalités de ce dernier peuvent être utilisées pour optimiser les performances du protocole de routage. En effet, à travers l'évaluation des comportements, les noeuds dont la valeur de réputation atteint un certain seuil de confiance seront considérés comme bienveillants, et la vérification de la signature numérique leur correspondant n'est plus nécessaire, ce qui va réduire considérablement le coût calculatoire engendré par ces opérations cryptographiques, et par conséquent, le rendement de chaque noeud sera amélioré.

A la fin de ce travail de recherche, nous pouvons dire que la sécurisation des protocoles de routage dans les réseaux ad hoc reste un vrai challenge. Les recherches continuent dans ce domaine afin d'améliorer et d'optimiser de plus en plus les solutions de sécurité existantes afin de rendre les réseaux ad hoc plus fiables, plus performants et plus sécurisés à faible coût pour le grand public.

Bibliographie

[1] Tounsi. N, Assassi. R, « **Simulation d'un protocole à base des colonies d'abeilles pour la qualité de service dans la couche réseau** », mémoire de fin d'études diplôme ingénieur d'état en informatique Université Mohammed Khider de Biskra, 2010,

[2] Ayad.k, « **Sécurité du routage dans les réseaux ad hoc mobile** », Thèse de magister en informatique, Ecole nationale Supérieure en Informatique (ESI), année 2011/2012.

[3] Abdelaziz, « **Approche agent mobile pour l'adaptation des réseaux mobiles ad hoc** », Présenté en vue de l'obtention du diplôme de Magister en Informatique, Université Mohamed Khider Biskra.

[4] Paul.M, « **802.11 et les réseaux sans fil** », livre Edition Eyrolles, 2002, ISNB : 2-212-11154-1.

[5] Boukhechem. N, « **Routage dans les réseaux mobiles Ad hoc par une approche a base d'agent** », mémoire Présenté en vue de l'obtention du diplôme de Magister en informatique, Université de Constantine, Promotion 2007-2008.

[6] Laouiti.A, « **Unicast et Multicast dans les réseaux Ad hoc** », thèse doctorat Université de Versailles Saint Quentin en Yvelines, Juillet 2002.

[7] Vander Meerschen Jérôme, « **Hybridation entre les modes Ad hoc et infrastructure dans les réseaux de type Wi-Fi** », mémoire de fin d'études, année 2006, Université Libre de Bruxelles.

[8] Sara .M, Nesrine Rym. G, **« Routage sécurisé des données dans les réseaux Ad hoc** ». Mémoire d'ingénieur d'état en informatique, ESI, Ecole Nationale Supérieure d'Informatique d'Oued-Smar, Alger. 2008/2009

[9] ABBES. Mounir. T, « **proposition d'un protocole à économie d'énergie dans un réseau hybride GSM et Ad hoc** ». Mémoire pour obtenir le diplôme de Doctorat spécialité informatique, université d'Oran 2011/2012.

[10] Mansouri. N, « **Protocole de routage multi chemin avec équilibrage de charge dans les réseaux mobiles Ad hoc** », Ecole supérieur des communications de Tunis, Tunisie, 2006-2007.

[11] Meskauskas. P, «**Mobile Ad hoc networking Seminar on Telecommunications Technology**», Helsinki, 1998.

[12] Telli. A, « **Algorithme de routage avec prise en compte de la consommation d'énergie dans les réseaux Ad hoc** », mémoire en vue de l'obtention du diplôme de Magister en Informatique, Université Mohammed Khider de Biskra, année 2010.

[13] Fabien.R, Nicolas.G, « **Le routage au sein des réseaux ad hoc** », Projet bibliographique, année 2004.

[14] Badache.N, Lemlouma.T, « **Le Routage dans les Réseaux Mobiles Ad Hoc** ».

[15] Marti. S, T.J. Giuli, K.Lai, and M.Baker. « **Mitigating Routing Misbehavior in Mobile Ad Hoc Networks**». ACM MOBIC, Boston MA, USA, pages 255-265, 2000.

[16] Maamar.Se, Lamia.A, G. Leila, B. Azeddine, « **Etude des Performances des Protocoles de Routage dans les Réseaux Mobiles Adhoc** » Département d'informatique, International Conférence on Computer Integrated Manufacturing CI P'2007, Université El Hadj Lakhdar – Batna.

[17] Exposé « **Les protocoles de routage hybrides** » université de Lill 1.

[18] Abdellaoui, R, « **SU-OLSR UNE NOUVELLE SOLUTION POUR LA SÉCURITÉ DU PROTOCOLE OLSR** » à l'obtention de la maitrise en génie concentration réseaux de télécommunications.

[19] Abdelmajid, H « **Sécurité du routage dans les réseaux sans fil spontanés : cas du protocole OLSR** » pour obtenir le grade de Docteur en Sciences Appliquées Spécialité Informatique.

[20] Aliouane, L, Badache N, « **L'Authentification dans les Réseaux Ad Hoc** », CERIST. RIST Vol, 16 n°01. 2006.

[21] BEGHRICHE, A, « **De la Sécurité à la E-Confiance basée sur la Cryptographie à Seuil dans les Réseaux sans fil Ad hoc Ad Hoc** », Université de Le Hadj Lakhdar-Batna, 2009.

[22] Samuel Galice, « **Modèle de sécurité dynamique pour les réseaux spontanés** »,

Institut National des Sciences Appliquées de Lyon. 2007.

[23] Étude technique réalisée par CGI, « **Étude technique Cryptographie à clé publique et signature numérique Principes de fonctionnement** », Septembre 2002.

[24] T. Clausen, P. Jacquet, « **Optimized Link State Routing Protocol (OLSR)**», Project Hipercom, INRIA in October 2003.

[25] Espès.D, « **Protocoles de routage réactifs pour l'optimisation de bande passante et la garantie de délai dans les réseaux Ad hoc mobiles** » En vue de l'obtention du Doctorat en informatique, l'université de Toulouse, année 2008

[26] https://fr.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol.

[27] Andreas Hafslund, Andreas Tonnesen, Roar B. Rotvik, Jon Andersson, and Oivind Kure « **Secure extension to the OLSR protocol. In OLSR Interop and Workshop** » August 2004.

[28] Daniele Raffo, Cédric Adjih, Thomas Clausen, and Paul Mûhlethaler « **An advanced signature system for OLSR** » ,2004.

[29] Amir R. Khakpour, Maryline Laurent-Maknavicius, and Hakima Chaouchi, « **WATCHMAN: An Overlay Distributed AAA Architecture for Mobile Ad hoc Networks** », IEEE Computer Society 2008.

[30] Vilela, J. P et J. Barros. **« A feedback reputation mechanism to secure the optimized link state routing protocol** », In Proceedings of the third International Conference on Security and Privacy in Communications Networks, 2007

[31] Hu, Y.-C., Perrig, A., and Johnson, D. B. **« Packet leashes: A defense against wormhole attacks in wireless networks** », In Proceedings of INFOCOM, Twenty-Second Annual Joint Conference of the IEEE Computer and Communication Societies (April 2003), vol. 3, pp. 1976_1986.

Annex

Le script OTCL de la configuration du réseau de la simulation

```
set val(chan)       Channel/WirelessChannel          ; # channel type
set val(prop)       Propagation/TwoRayGround         ; # radio-propagation model
set val(netif)      Phy/WirelessPhy                  ; # network interfacce type
set val(mac)        Mac/802_11                       ; # Mac type
set val(ifq)        CMUPriQueue                      ; # interface queue type
set val(ll)         LL                               ; # Link layer type
set val(ant)        Antenna/OmniAntenna              ; # antenna Model
set val(ifqlen)     50                               ; # max packeet in ifq
set val(nn)         20                               ; # number of mobilenode
set val(rp)         OLSR                             ; # routing protocol
set val(x)          1000                             ; # x dimension of topography
set val(y)          500                              ; # y dimention of topography
set val(stop)       300                              ; # time of simulation end

#-------Event scheduler object creation--------#

    set ns [new Simulator]
## Create a trace file and nam file..
    set tracefd [open wireless2.tr w]
    set namtrace [open wireless2.nam w]

## Trace the nam and trace details from the main simulation..
    $ns trace-all $tracefd
    $ns namtrace-all-wireless $namtrace $val(x) $val(y)
## set up topography object..
    set topo [new Topography]

    $topo load_flatgrid $val(x) $val(y)
    set god_ [create-god $val(nn)]

## Setting node config event with set of inputs..

      $ns node-config -adhocRouting $val(rp) \
              -llType $val(ll) \
              -macType $val(mac) \
              -ifqType $val(ifq) \
              -ifqLen $val(ifqlen) \
              -antType $val(ant) \
              -propType $val(prop) \
              -phyType $val(netif) \
              -channelType $val(chan) \
              -topoInstance $topo \
              -agentTrace ON \
              -routerTrace ON \
              -macTrace OFF \
              -movementTrace ON
```

```
#Set a UDP connection between nodes
set udp [new Agent/UDP]
#$udp set class_ 2
set null [new Agent/Null]
$ns attach-agent $node_(0) $udp
$ns attach-agent $node_(1) $null
$ns connect $udp $null
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$ns color 1 Blue
$ns at 1.0  "$cbr start"
$ns at 700.0 "$cbr stop"
## Define node initial position in nam..
      for {set i 0} {$i < $val(nn)} { incr i } {
      # 30 defines the node size for nam..
            $ns initial_node_pos $node_($i) 30
      }


$ns at $val(stop) "stop"

#stop procedure
proc stop {} {
    global ns tracefd namtrace
    $ns flush-trace
    close $tracefd
    close $namtrace
    puts "running nam..."
    exec nam wireless2.nam &
    exit 0
}

$ns run
```

#snapshot of the program