الجمـهورية الجزائرية الديمقراطية الشعبيـة
وزارة التعليم العالي والبحث العلمي

جـامعـة سعـيـدة د. مـولاي الطـاهـر
كـلـيـة التكنولوجيا
قسم: الإعلام الآلي

MÉMOIRE DE MASTER

SPÉCIALITÉ : SÉCURITÉ INFORMATIQUE ET CRYPTOGRAPHIE

Thème

Sécurité de protocole de routage Adhoc

Présenté par :

- Semmani Mohammed
- Nouari Taqiyeddine

Dirigé par :

- Mr. Henoune Mohammed Mokhtar

Promotion 2021 - 2022

Dédicaces

A Mes chers parents,
A la mémoire de mon grand-père,
A mes frères et mes sœurs,
A cher ami « Djemal-eddine » ,
A mon professeur « Louakel Mohammed » ,
A tous mes amis (es),

« Mohammed »

A Mes très chers parents,
A mes frères,
A mes chers amis « Redhwane Maddi et Oussama » ,
A tous mes amis (es),

« Taqiyeddine »

Remerciements

Nous tenons à exprimer notre profonde gratitude à notre promoteur, Monsieur « Henoune Mohammed Mokhtar »

Pour nous avoir encadrés durant cette année, ainsi que pour ses conseils judicieux.
Nos remerciements vont également aux membres du jury pour l'honneur qu'ils nous
font en acceptant d'examiner et de juger notre travail.
Nous remercions aussi tous ceux, et celles qui ont contribué de près ou de loin pour
l'accomplissement de ce modeste travail.

ملخص.

الشبكة اللاسلكية المخصصة هي عبارة عن مجموعة من العقد المتنقلة التي تشكل شبكة مؤقتة ذات طوبولوجيا متغيرة وتعمل بدون محطة أساسية وبدون إدارة مركزية، يمكن الاتصال متعدد القفزات بفضل بروتوكولات توجيه محددة. في هذه الدراسة، سنقوم بتأمين بروتوكول توجيه AODV عن طريق استعمال التطبيق المحاكي OMNET++.

لحماية طوبولوجيا الشبكة وتأمين المسارات بين العقد، اقترحنا استخدام IPSec الذي يسمح بضمان حماية المسارات على الشبكة. وسنستخدم على وجه التحديد بروتوكول AH الأساسي (ترويسة المصادقة) في وضع النقل الذي يوفر سلامة ومصادقة حزم التوجيه.

الكلمات المفتاحية : توجيه Adhoc، AODV، حزم التحكم، OMNET، الحماية ، IPSEC، AH.

Abstract

A wireless ad-hoc network is a collection of mobile nodes forming a temporary network with variable topology and operating without a base station and without centralized administration, multi-hop communications are possible thanks to specific routing protocols. In this study we will contribute to secure the AODV routing protocol by simulation under OMNET ++.

To protect the network topology and secure the paths between the nodes we suggested to use IPSec which allows to ensure the protection of the routes on the network. And precisely we will use its basic AH (Authentication header) protocol in transport mode which provides integrity and authentication of routing packets.

Key words: Adhoc routing, AODV, Control packets, OMNET, Security, IPSEC, AH.

Résumé

Un réseau ad-hoc sans fil est une collection de nœuds mobiles formant un réseau temporaire à topologie variable et fonctionnant sans station de base et sans administration centralisée, les communications multi sauts y sont possibles grâce à des protocoles de routage spécifiques. Dans cette étude nous contribuerons à sécuriser le protocole de routage AODV par la simulation sous OMNET ++.

Pour protéger la topologie du réseau et sécuriser les chemins entre les nœuds nous avons suggéré d'utiliser d'IPSec qui permet d'assurer la protection des routes sur le réseau. Et précisément on va utiliser son protocole de base AH (Authentication header) en mode transport qui fournit l'intégrité et l'authentification des paquets de routage.

Mots clés : routage Adhoc, AODV, Paquets de contrôle, OMNET, Sécurité, IPSEC, AH.

Sommaire :

Table des figures :

Liste des Tableaux:

Glossaire :

MANET : Mobile Ad hoc Networks

OSI : Open Systems Interconnection.

TTL , Time To Live

DSR, Dynamic Source Routing.

DSDV, Destination-Sequenced Distance Vector.

AODV, Ad-hoc On Demand Distance Vector.

TORA, Temporally-Ordered Routing Algorithm.

OLSR Optimized Link State Routing Protocol.

TBRPF, Topology Broadcast Based on Reverse-Path Forwarding.

IPsec : Internet Protocol Security

AH : authentication header

ESP : Encapsulating Security Payload

CA : Certificate Authorities

PFS : Perfect Forward Secrecy

SAD : Security Association Data base.

SPD : Security Political Data base.

VPN : Virtual Private Network

IKE : internet key exchange

PKI : Public Key infrastructure

PFS : Perfect Forward Secrecy .

ISAKMP : internet security association and key management protocole

OMNeT : Objective Modular Network Test-bed

Introduction générale

Ces dernières années, l'évolution des technologies sans fil a ouvert de nouvelles perspectives dans le domaine des télécommunications. En particulier, les réseaux mobiles sans fil connaissent une forte expansion. Les réseaux mobiles offrent une grande flexibilité d'emploi en permettant aux utilisateurs de se déplacer librement tout en continuant leurs communications.

Les réseaux mobiles sans fil, peuvent être classés en deux classes : les réseaux avec infrastructure ou cellulaire et les réseaux sans infrastructure (Ad-Hoc). Plusieurs systèmes utilisent le modèle cellulaire et connaissent un très fort épanouissement à l'heure actuelle, mais requièrent une importante infrastructure matérielle fixe.

Un réseau sans fil ad hoc, appelé généralement MANET (Mobile Ad hoc NETWORK), est un système autonome composé par un ensemble d'entités mobiles utilisant le médium radio pour communiquer. Il s'auto-organise et opère sans recourir à une infrastructure préexistante ou une administration centralisée. Dans un tel réseau, les nœuds qui sont hors-portée radio les uns des autres comptent sur la coopération des nœuds intermédiaires pour acheminer les données à la destination. Bien que simple, rapide et moins coûteux à déployer, un réseau sans fil ad hoc est vulnérable par plusieurs types d'attaques. En effet, à cause de l'ouverture du médium de communication, la mobilité et l'absence d'infrastructure, un nœud malveillant peut facilement écouter, modifier ou supprimer le trafic passant par lui. Il peut aussi cesser d'acheminer les données, tandis qu'il sollicite les autres nœuds pour lui acheminer ses données. Afin de prévenir ce type d'action des nouds malveillant, il est primordial de protéger le protocole de routage en appliquant des mécanismes de sécurité aux paquets de contrôle (routage). En ce qui concerne la protection des données il existe plusieurs protocoles de sécurité de bout en bout qui peuvent être configure par l'utilisateur [1].

Dans ce mémoire on vise à protéger le protocole de routage AODV en appliquant le protocole de sécurité IPSEC sur les paquets de contrôle. Cette intégration des mécanismes de sécurité a un impact sur les performances de protocole de routage en termes de charge, latence, etc…. Notre but est d'étudier cet impact est voir quelle sont les avantages et les inconvénients de notre approche.

Dans un premier temps nous allons parler des réseaux ad hoc ainsi les protocoles de routage ad hoc. Ensuite on détaille le protocole de routage AODV. Dans le troisième chapitre on explique le fonctionnement de protocole IPSEC. Dans le dernier chapitre en va proposer une approche qui sécurise le protocole de routage avec une étude sur ses effets sur les performances de protocole de routage.

Chapitre I : Généralisation sur les réseaux Ad hoc

I.1 Introduction :

Un réseau ad hoc sans fil est un système composé de nœuds (ordinateur portable, PDA, Net book, etc.) éventuellement mobiles, qui permet à ses utilisateurs de communiquer via des ondes radio [1]. Deux types de réseaux sans fil peuvent être distingués :

Le réseau avec infrastructure qui est constitué de plusieurs stations de bases, reliées entre elles par une architecture filaire jouant le rôle d'un routeur pour faire communiquer des nœuds assignés à des stations de bases différentes.

Un réseau Ad Hoc est un ensemble des nœuds mobile, l'acheminement de l'information d'une source vers une destination nécessitent des protocoles de routage établissent des routes entre les nœuds de réseau.

Il s'agit de réseaux sans fil composé d'un ensemble relativement dense des 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 son porté 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 (mobiles) de réseau.

Dans ce chapitre, nous allons présenter les principaux concepts liés au réseau sans fil Ad hoc. Nous commencerons par la définition de ce type de réseaux et son domaine d'application ainsi que quelques caractéristiques, puis nous allons définir le routage dans le réseau ad hoc

I.2 Les environnements mobiles :

Un environnement mobile est un système composé de sites mobiles et qui permet à ses utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques [2].

Nous pouvons distinguer deux classes de réseaux mobiles, à savoir **(Figure I.1)** :

- Les réseaux basés sur une infrastructure de communication (modèle cellulaire)
- Les réseaux mobiles sans infrastructure (modèle ad hoc)

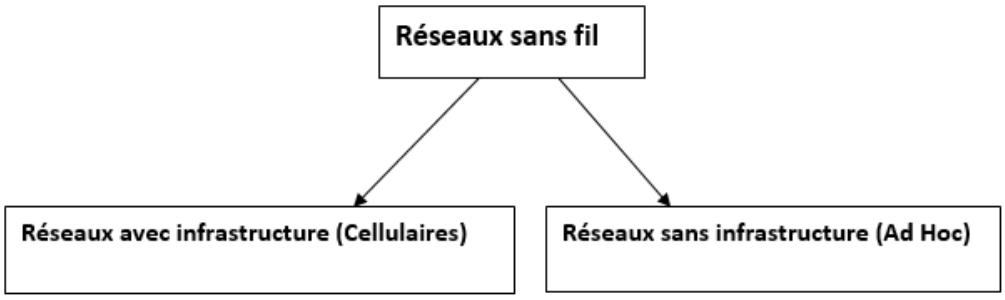


Figure I.1: Les Types de réseaux sans fil

I.2.1 Réseaux avec infrastructure :

Ces réseaux sont constitués de deux ensembles d'entités **(Figure I.2)**:

- Les "sites fixes" d'un réseau de communication filaire appelés "stations de bases"
- Les "sites mobiles" appelés hôtes ou nœuds mobiles [3].

Chaque station de base est munie d'une interface de communication sans fil pour la communication directe avec les nœuds mobiles situés dans sa zone de couverture dite cellule. Les hôtes mobiles communiquent entre eux via le réseau des stations de base. Notons qu'à un instant donné, un nœud mobile ne peut être connecté qu'à une seule station de base.

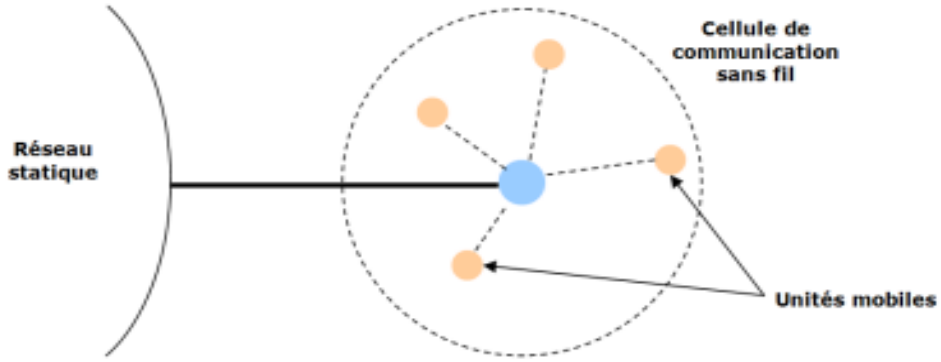


Figure I.2: Le modèle des réseaux mobiles avec infrastructure.

I.2.2 Réseaux sans infrastructure :

Les réseaux sans infrastructure ou ad hoc sont constitués d'un ensemble de nœuds mobiles qui communiquent entre eux sans l'aide d'une infrastructure préexistante ou d'une administration centralisée **(Figure I.3)**. Cela nécessite la coopération de tous les nœuds mobiles afin de garantir les fonctionnalités nécessaires du réseau comme le routage et l'accès au médium de communication. En réalité, dans un réseau ad hoc les unités mobiles jouent à la fois le rôle de terminaux et de routeurs afin d'acheminer l'information entre elles [4].

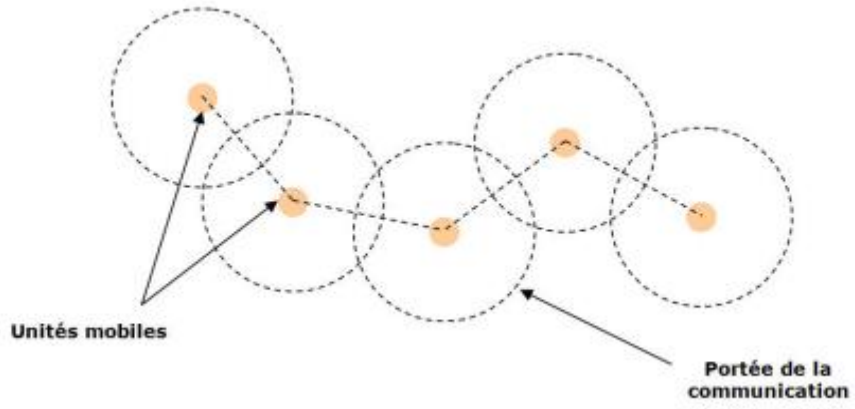


Figure I.3: Le modèle des réseaux mobiles sans infrastructure.

I.3 Les réseaux mobiles Ad Hoc :

I.3.1 Bref historique :

Le début des années 1970 voit, au sein du projet militaire Américain DARPA (The Defense Advanced Research Projects Agency), la naissance des premiers réseaux utilisant le médium radio. Ces réseaux disposaient déjà d'une architecture distribuée, partageaient le canal de diffusion en répétant des paquets pour élargir la zone de couverture globale. Par la suite, en 1983, les Survivable Radio Networks (SU RAN) furent développés par le DARPA. L'objectif était de dépasser les limitations (en particulier permettre le passage à des réseaux comportant énormément des nœuds, gérant la sécurité, l'énergie). Mais les recherches sur ces réseaux restaient exclusivement militaires. Ce n'est qu'avec l'arrivée du protocole 802.11 de l'IEEE (Institute of Electrical and Electronics Engineers) qui permet de bâtir des réseaux sans fil autour de bases fixes, que la recherche civile s'empare à la fin des années 90s des problématiques liées à ces réseaux.

I.3.2 Définition :

Un réseau mobile ad hoc est un environnement mobile et sans infrastructure, appelé généralement MANET (Mobile Ad hoc NETWORK) [5], consiste en une grande population, relativement dense, d'unités mobiles qui se déplacent dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée.

Les systèmes de communication cellulaire sont basés essentiellement sur l'utilisation des réseaux filaires (tel que les réseaux d'opérateurs ou ATM) et la présence des stations de base qui couvrent les différentes unités mobiles du système.

Les réseaux mobiles « ad hoc » sont à l'inverse, des réseaux qui s'organisent automatiquement de façon à être déployé rapidement, sans infrastructure fixe, et qui doivent pouvoir s'adapter aux conditions de propagation, aux trafics et aux différents mouvements pouvant intervenir au sein des nœuds mobiles

I.3.3 Définition de routage :

Le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. 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.).

Le routage est la tâche d'acheminement de flux des données à partir des nœuds sources vers les nœuds destinations. Si une seule destination est impliquée dans la communication, alors il s'agit d'un "routage unicast", si encore tous les nœuds du réseau ou juste un sous ensemble sont concernés par la réception des données alors on parle du :

- **La communication point a point « unicasting » :** Dans ce mode de communication le paquets est adressé à un seul nœud mobile.
- **La communication multi point « multicasting »** : contrairement à l'unicast, un paquet est adressé à un ensemble des unités mobile dans le réseau.
- **La diffusion Broadcast** : un paquet est adressé à toutes les unités composant le réseau,

La figure suivante présente les trois modes de communication citée précédemment : **(figure I.4).**

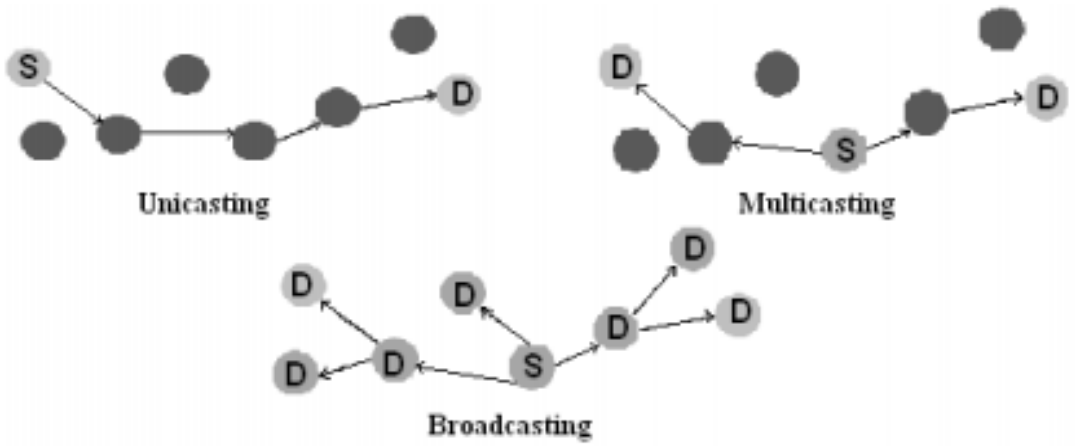


Figure I.4: Illustration du routage unicast, multicast et broadcaste.

I.4 Classification des Protocoles de routage :

Suivant la classification des protocoles de routage dans les réseaux ad hoc, les protocoles de routage peuvent être séparés en deux catégories, les protocoles proactifs et les protocoles réactifs. Les protocoles proactifs établissent les routes à l'avance en se basant sur l'échange périodique des tables de routage, alors que les protocoles réactifs cherchent les routes à la demande.

D'autres classes existent tel que : les protocoles de routage hybrides qui combinent les deux approches précédentes afin de tirer avantage de deux catégories citées précédemment, tout en réduisant leur limitation, on cite aussi les protocoles géographiques, hiérarchique, à qualité de service et multicast. [6] **(figure I.5)**

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent classer en deux familles de protocoles :

- **État des liens :** TORA, OLSR et TBRPF.
- **Vecteur de distance :** DSR, DSDV et AODV.

Suivant le moment de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en :

- **Proactif** : **DSDV**, **OLSR** et **TBRPF** adoptent ce comportement. Les protocoles proactifs établissent les routes à l'avance en se basant sur l'échange périodique des tables de routage.
- **Réactif** (sur demande) : **TORA** et **AODV** adoptent ce comportement. Les protocoles réactifs cherchent les routes à la demande. **AODV** est en fait une version réactive de DSDV.
- **Hybride** : les protocoles hybrides définissent deux zones où ils combinent le comportement proactif à l'intérieur d'une zone et le comportement réactif entre les zones. Par exemple **DSR**, qui est réactif à la base mais qui peut être optimisé s'il adopte un comportement proactif [7].

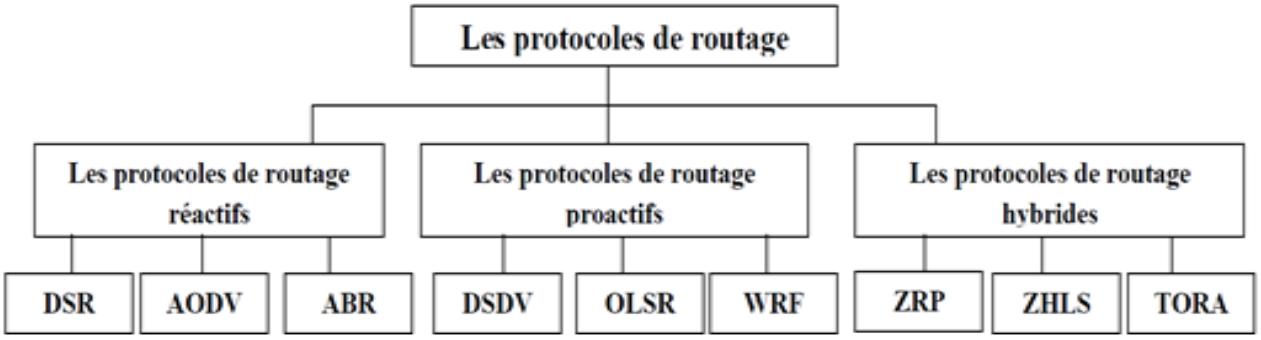


Figure I.5 : les déférents types des protocoles de routage

I.4.1 Les protocoles proactifs :

Les protocoles de cette catégorie sont basés sur les algorithmes classiques d'état de liens et de vecteur de distance déjà présentés dans la section précédente. Le principe de base de ces protocoles est de maintenir à jour les tables de routage, et ce par la mise en place d'un système d'échange périodique des paquets de contrôle. Cette manière de procéder permet aux nœuds de construire de façon distribuée la topologie du réseau.

Il existe, à cet effet, deux types de paquets de contrôle : les paquets envoyés localement (à un saut) pour la découverte du voisinage et les paquets diffusés dans tout le réseau pour communiquer aux autres nœuds les informations sur l'état du voisinage (généralement l'ensemble des voisins ou un sous ensemble) rassemblés par le premier type de messages de contrôle. 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 paquets de contrôle. Ce processus est déclenché aussi à chaque changement de topologie pour reconstruire à nouveau les routes. La caractéristique principale de ces protocoles est la disponibilité immédiate de la route lors du besoin. Les principaux protocoles proactifs sont : **OLSR** et **DSDV** [8]

I.4.1.1 Le Protocoles DSDV :

L'algorithme DSDV (Dynamic destination Sequenced DistanceVector) a été conçu spécialement pour les réseaux mobiles. Chaque station mobile maintient une table de routage qui contient toutes les destinations possibles, le nombre des sauts pour atteindre la destination,

Le numéro de séquences (SN) qui correspond à un nœud de destination, permettant de distinguer les nouvelles routes des anciennes et d'éviter la formation de boucles de routage.

Les mises à jour des tables sont transmises périodiquement à travers le réseau afin de maintenir la consistance des informations ce qui génère un trafic important qu'il faut limiter.

Pour cela, deux types de paquets de mise à jour sont utilisés : les "fulls dump", contenant toutes les informations et des paquets plus petits, ne contenant que les informations ayant changé depuis le dernier "full dump" [9].

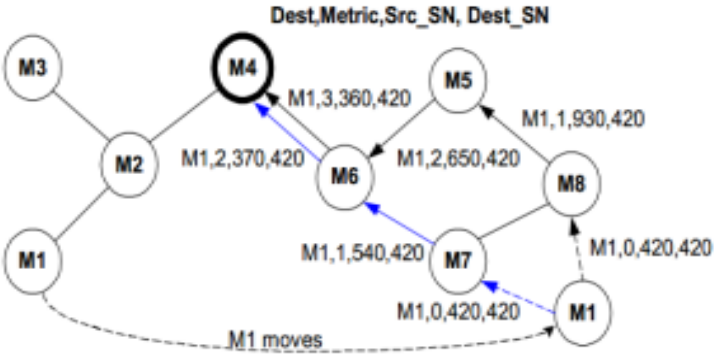


Table de routage de M4 :

Avant déplacement de M1

Destination	NextHop	Metric	Dest_SN
M1	M2	2	410
M2	M2	1	780
M3	M2	2	250

Après déplacement de M1

Destination	NextHop	Metric	Dest_SN
M1	M6	3	420
M2	M2	1	780
M3	M2	2	260

Figure I.6: Exemple d'échange DSDV

I.4.1.2 Le Protocoles OLSR :

OLSR est un protocole proactif basé sur un algorithme de type « êtas des liens ». Pour éviter l'inondation classique (flooding) sur ce type d'algorithme, le protocole prévoit l'élection de nœud spécifique les MPR (Multipoint Relay) sont chargé de transmettre les informations de topologie.

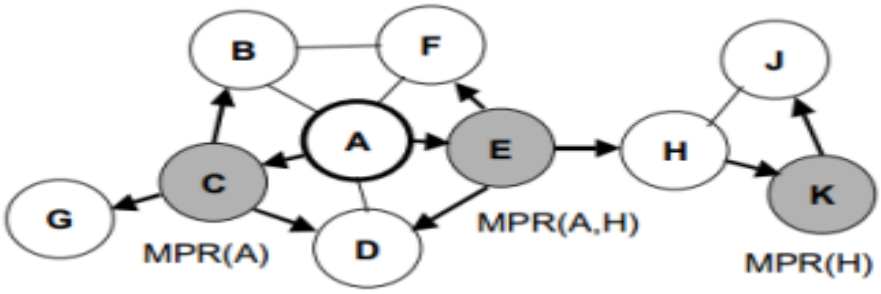


Figure I.7: Exemple échange OLSR

OLSR présente l'avantage de réduire la taille des messages de contrôle, car seuls les liens aux MPRs sont déclarés. En plus, l'inondation est limitée par l'utilisation des MPRs.

I.4.1.3 Avantages et les inconvénients des protocoles proactifs :

Avec un protocole proactif, les routes sont disponibles immédiatement, ainsi l'avantage d'un tel protocole est le gain de temps lors d'une demande de route. Le problème est que, les changements de routes peuvent être plus fréquents que la demande de la route et le trafic induit par les messages de contrôle et de mise à jour des tables de routage peut être important et partiellement inutile, ce qui gaspille la capacité du réseau. De plus la taille des tables de routage croit linéairement en fonction du nombre de nœud.

I.4.2 Les protocoles réactifs :

Les protocoles de routage proactifs 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.

L'idée de cette approche est de lancer le processus de recherche de routes uniquement au besoin (à la demande). Ainsi, quand un nœud demande de relayer ses données vers une destination quelconque dont il ne dispose pas de route valide, le mécanisme de recherche de route est déclenché. Le principe est le suivant :

- Le nœud émetteur diffuse une requête de route « Route Request » au niveau de son rayon de propagation.
- Le mécanisme d'inondation permet à cette requête de se propager sur tout le réseau.
- À chaque fois le paquet passe par un nœud mobile, jouant dans ce cas le rôle d'un routeur, une information portant l'identifiant du nœud est ajoutée à la route jusqu'à ce que le paquet atteigne sa destination.
- À la réception du paquet, le nœud destinataire renvoie le tracé du chemin à la source en suivant le chemin inverse.
- Parmi toutes les routes renvoyées par le nœud destinataire, le nœud source sélectionne la plus optimale et il la sauvegarde dans sa cache (table de routage) afin de l'utiliser immédiatement ou pour ces besoins futurs.

Cependant, du fait que l'on ne dispose pas immédiatement de la route vers la destination, le délai nécessaire à l'acheminement des paquets vers la destination est plus important en comparaison avec les protocoles proactifs.

Les principaux protocoles réactifs sont : AODV et DSR [9].

I.4.2.1 Le Protocoles DSR :

DSR (Dynamic Source Routing) est basé sur l'utilisation du technique "routage à la source" c'est- à-dire c'est à la source de déterminer la séquence complète des nœuds selon lesquelles, les paquets de données seront envoyés. Les nœuds n'ont pas besoin de tables de routage. Les deux opérations de base de DSR sont : la découverte de routes (route discovery) et la maintenance de routes (route maintenance). La découverte de routes se fait par diffusion d'un paquet requête de route pour identifier la cible. En cas de réussite, le nœud initiateur reçoit un paquet réponse de route qui liste la séquence de nœuds à travers lesquels la destination peut être atteinte. Dès que la destination est localisée, une copie de ce chemin est envoyée dans un paquet réponse de route à l'initiateur. De cette manière, la requête de route est propagée dans le réseau, jusqu'à ce qu'elle atteigne la destination qui va répondre à la source.

Afin de réduire le coût et la fréquence de la découverte de routes, chaque nœud garde trace des chemins trouvés à l'aide des paquets de réponses. Ces chemins sont utilisés jusqu'à ce qu'ils soient invalides. Le protocole DSR n'intègre pas l'opération de découverte de routes avec celle de la maintenance, comme le fait les protocoles de routage conventionnels et évite le problème de boucle de routage.

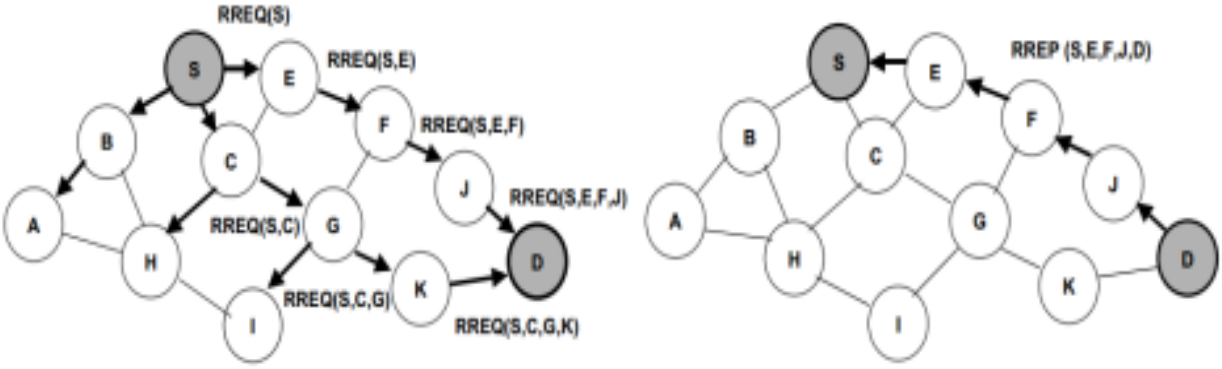


Figure I.8: Exemple de découverte DSR

I.4.2.2 Le protocole AODV :

AODV «Ad hoc On-Denzand Distance Vector» [10] a été conçu comme une amélioration de DSDV. Il s'agit d'un protocole de routage réactif, qui utilise le routage par saut : chaque noeud qui reçoit un paquet a pour charge de déterminer à qui l'envoyer en consultant sa table de routage.

Comme dans DSR, le noeud source diffuse une requête de route (RREQ) pour obtenir un chemin vers une destination donnée. Cette requête est diffusée dans le cas où le noeud source n'a pas de route vers la destination ou que la durée de vie de la route a expiré. Chaque requête de route est identifiée par un numéro de requête qui, avec l'adresse du noeud source, identifie d'une façon unique la requête dans le réseau. Pendant la propagation de la requête de route, les noeuds intermédiaires enregistrent dans leurs tables de routage l'adresse du noeud qui leur a envoyé la requête dans l'entrée (dans la table de routage) correspondant au nœud source afin d'établir un chemin inverse pour cette requête, voir la **(figure I.9)** pour une ilLustration.

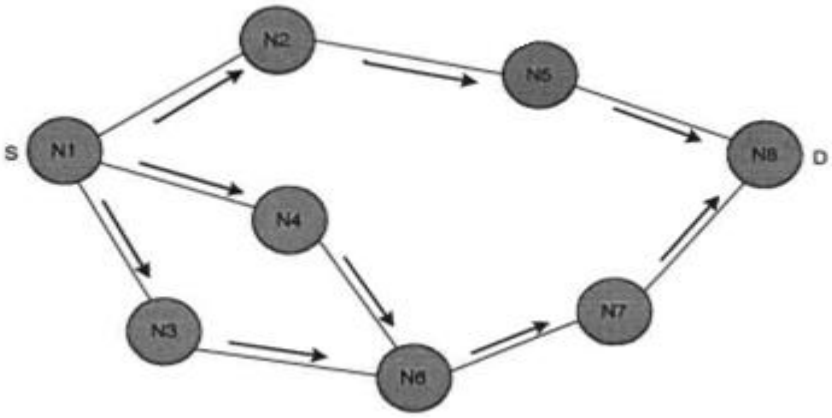


Figure I.9: propagation du paquet RREQ dans AODV

Une fois la requête arrivée au noeud destination, ce dernier répond par un paquet réponse de route (RREP). Pendant la propagation du paquet RREP les nœuds intermédiaires mettent à jour l'entrée correspondant au noeud destination dans leurs tables de routage, voir la **(figure I.10)**

Compte tenu du fait que les noeuds mettent à jour leurs tables de routages en établissant des chemins inverses, AODV fonctionne sur les réseaux ayant des liens symétriques.

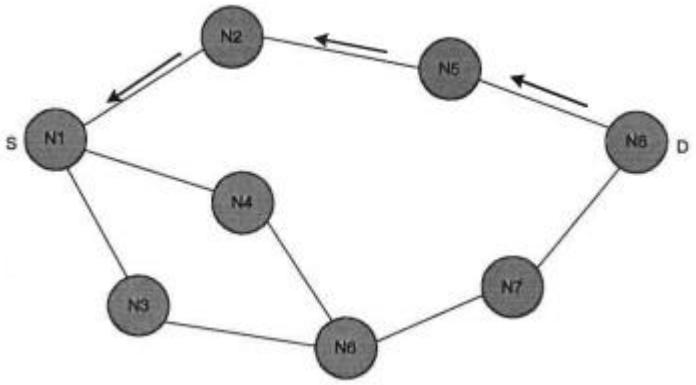


Figure I.10: propagation du paquet RREP dans AODV

Plusieurs autres protocoles de routage réactif ont été proposés pour les réseaux ad hoc [9].

I.4.2.3 Avantages et les inconvénients des protocoles réactifs :

A l'opposé des protocoles proactifs, dans les protocoles réactifs aucun message de contrôle ne change le réseau pour des routes inutilisées ce qui permit de préserver les ressources du réseau (bande passante).

Mais la mise en place d'une route par inondation peut être couteuse, et provoque des délais importants avant l'ouverture de la route et les retards dépassent bien souvent les délais moyens admis par les logiciels, ce qui provoque une erreur et impossibilité de se connecter alors que le nœud destination est la et la route existe.

I.4.3 Protocoles de routage Hybrides

C'est une combinaison des deux concepts de routage proactif et réactif. Des tables de routage sont stockées sur les nœuds maitres de façon à établir des routes sur leur voisinage proche (généralement en deux sauts maximums). Au-delà de leur voisinage, le routage devient réactif et des procédures de recherche de routes sont lancées. Cette approche combine les avantages des deux autres approches proactive et réactive et réduit considérablement la taille des tables de routage ainsi que les délais d'établissement de routes.

I.4.3.1 Le Protocoles ZRP

Le protocole ZRP (Zone Routing Protocol) est un exemple de protocole hybride, à mi-chemin entre les deux familles de protocoles. Ainsi, chaque nœud maintient une table de routage dont les données sont régulièrement émises en diffusion pour tous les nœuds qui lui sont distants de moins qu'une valeur d prédéfinie (routage pro actif dans cette zone). Pour atteindre tout autre nœud qui n'apparaîtrait pas dans sa table de routage (donc distant de plus de la distance d), un nœud a recours à un protocole de routage de type réactif similaire au protocole DSR. Ce type de protocole fournit un assez bon compromis en termes de diffusion pour les mises à jour. Cette tentative pour cumuler les qualités des deux approches, bien que notable, se place en intermédiaire plus qu'en solution, parce qu'elle est moins efficace en forte mobilité ou avec beaucoup de stations que les algorithmes de routage de base. Il combine à la fois :[11]

- Une approche proactive à l'intérieur d'une zone restreinte, qui permet de mettre à jour l'état du réseau et de maintenir des routes qu'il y ait ou non des paquets de données circulent.

- Une approche réactive entre les zones restreintes, qui ne détermine une route entre des nœuds périphériques que si le besoin de transmettre des paquets de donné apparait

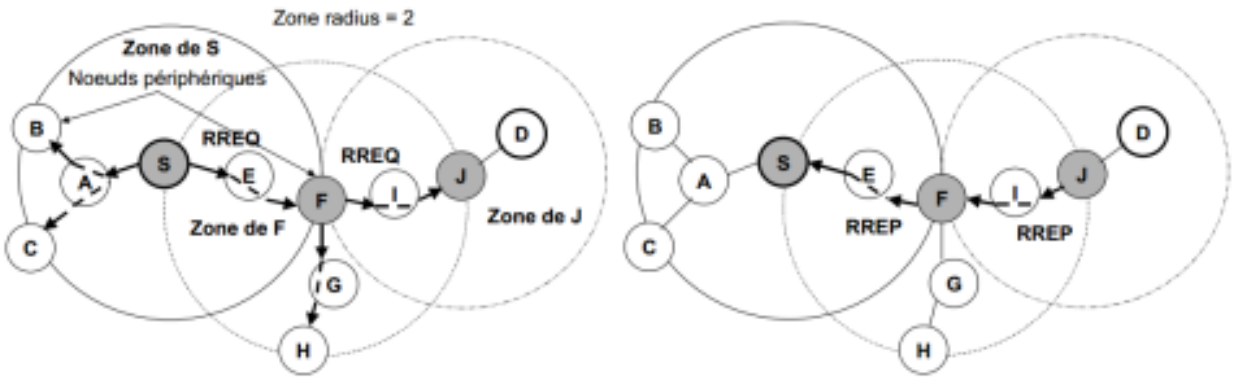


Figure I.11: Exemple de routage ZRP

I.4.3.2 Le Protocoles ZHLS :

Le protocole ZHLS est basé sur la décomposition d'un réseau en zones. Contrairement à la plupart des protocoles dit hiérarchiques, il n'y a pas ici de représentant pour chaque zone.

La topologie d'un réseau est ainsi partagée en deux niveaux (**Figure I.12)**:

✓ Un niveau nœud indique la façon dont les nœuds d'une zone sont connectés entre eux physiquement. Un lien virtuel peut exister entre deux zones s'il existe au moins un nœud d'une autre zone.

✓ Un niveau zone qui renseigne sur le schéma de connexion des différentes zones.

Ces niveaux différents entraînent donc deux différents types de liens : les liens inter-nœuds et les liens interzones. Le réseau est donc décomposé comme l'illustre la figure. Il résulte de cette décomposition un routage interzone et un routage intra-zone qui est permise par l'adressage mis en place et qui consiste en un identifiant de zone, un identifiant de nœuds et l'utilisation de LSP (Link State Packet) qui renseignent sur l'état des liens. Il est alors également possible de distinguer deux classes de LSP : la classe des LSP orientés nœuds pour lesquels un nœud donné contient des informations sur son voisin et celle des LSP orientés zones qui sont, quant à elles, échangées de manière globale. Ainsi chaque nœud du réseau possède une connaissance complète concernant les nœuds de sa propre zone et seulement une connaissance partielle du reste des nœuds. Les nœuds déterminent leur position physique en utilisant le GPS. La carte de zone est établie pendant la phase de composition du réseau.

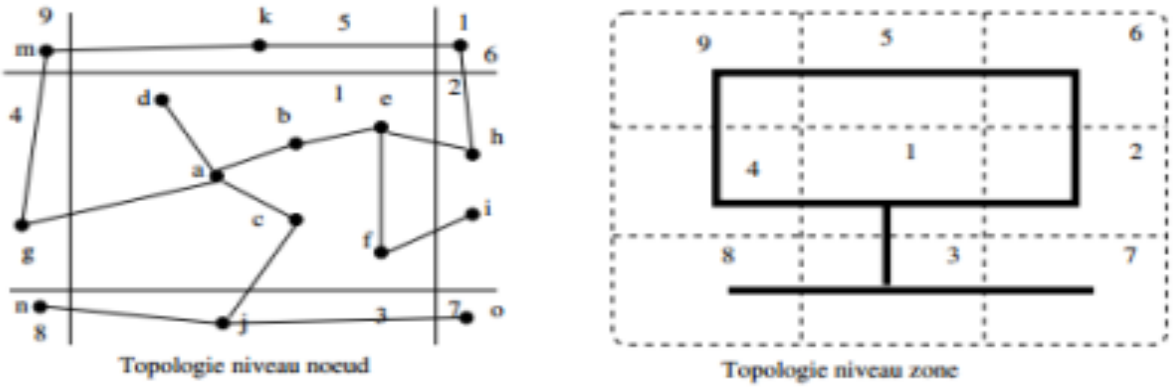


Figure I.12 : protocole hybride ZHLS

I.4.3.3 Avantages et les inconvénients des protocoles hybrides :

Le protocole hybride est un protocole qui se veut comme une solution mettant en commun les avantages des deux approhes précédentes en utilisant une notion de découpe du réseau.

Cependant, il rassemble toujours quelques inconvénients des deux approches proactives et réactives.

I.5 Conclusion :

Après avoir défini l'environnement mobile Ad hoc et décrit ses caractéristiques, nous avons parlé du routage et la classification des protocoles de routage dans les réseaux mobiles Ad Hoc.

A cause des caractéristiques inhérentes de ces réseaux, un vrai problème de sécurité se pose. Dans le chapitre suivant, nous allons d'abord examiner en détail le protocole de routage réactif AODV et après présenter les différentes attaques possibles contre ce protocole de routage.

Chapitre II : Le protocole de routage AODV

II.1 Introduction :

Plusieurs protocoles de routage ont été proposés pour le contexte ad hoc, qu'ils s'agissent d'adaptation des protocoles de routage Internet ou de nouveaux protocoles spécifiques aux réseaux MANET. Le principal but de toutes stratégies de routage est de mettre en œuvre une bonne gestion d'acheminement qui soit robuste et efficace. D'une manière générale, toutes stratégies de routage, reposent sur des méthodes et des mécanismes que l'on peut regrouper en trois grandes classes : les protocoles de routage proactifs, les protocoles de routage réactifs.
Le protocole AODV fait le sujet principal de ce chapitre, nous définissons d'abord le protocole de routage réactif AODV, on montre les paquets de contrôle utilisés par le protocole ainsi que sa table de routage et son mécanisme de fonctionnement en tant que découverte de route et maintenance des routes. Ensuite, nous présentons les différentes attaques possibles contre ce protocole de routage.

II.2 Définition de routage dans les réseaux ad hoc :

Le routage est un processus qui permet de sélectionner un et/ou plusieurs chemins dans un réseau pour transmettre des données depuis un expéditeur jusqu'à un ou plusieurs destinataires. On parle de routage dans différents domaines : Réseaux téléphoniques et réseaux de transports. Sa performance est importante dans les réseaux décentralisés, c'est-à-dire où l'information n'est pas distribuée par une source unique, mais échangée entre des mobiles indépendants [12].

Le terme routage désigne l'ensemble des mécanismes mis en œuvre dans un réseau pour déterminer les routes qui vont acheminer les paquets d'un terminal émetteur à un terminal récepteur [13].

II.3 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. Ces protocoles se basent sur la découverte et le maintien des routes. Suite à un besoin, une procédure de découverte globale de routes est lancée. Ce processus s'arrête une fois la route trouvée ou toutes les possibilités sont examinées. Dès que la communication est établie, cette route est maintenue jusqu'à ce que la destination devienne inaccessible ou jusqu'à ce que la route ne soit plus désirée. Parmi les protocoles basés sur ce principe on cite : DSR, AODV, ABR, LAR ...etc [14].

II.4 Le Protocole AODV :

AODV (Ad hoc On demand Distance Vector) est un protocole de routage conçu par Charles E.Perkins et Elizabeth M. Royer [15]. C'est un protocole basé sur le principe des vecteurs de distance et appartient à la famille des protocoles réactifs. Il représente essentiellement une amélioration de l'algorithme proactif DSDV mais réduit le nombre de diffusions de messages en ne calculant les routes que sur demande (AODV). Ce protocole utilise les deux mécanismes "découverte de route" et "maintenance de route", en plus du routage "nœud par nœud" ; le principe des « numéros de séquence » et « l'échange périodique des paquets HELLO » sont inspirée du DSDV.

AODV utilise le principe des numéros de séquence afin de maintenir la consistance des informations de routage. A cause de la mobilité des nœuds dans les réseaux mobiles ad hoc, les routes changent fréquemment ce qui fait que les routes maintenues par certains nœuds, deviennent invalides. Les

numéros de séquence permettent d'utiliser les routes les plus nouvelles ou autrement dit les plus fraîches [16].

II.4.1 Une table de routage contient :

- L'adresse de la destination.
- Le nœud suivant.
- La distance en nombre de nœud.
- Le numéro de séquence destination.
- Le temps d'expiration de l'entrée de la table.

II.4.2 Le protocole AODV fonctionne à partir de trois types de messages :

- Les messages de demande de route RREQ : Route Request Message.
- Les messages de réponse de route RREP : Route Reply Message.
- Les messages d'erreur de route RERR : Route Error Message.

L'AODV utilise le principe de numéro de séquence afin d'éviter le problème des boucles infinie et des transmissions inutiles de messages sur le réseau, en plus il permet de maintenir la consistance des informations de routage ainsi les numéros de séquence permettent d'utiliser les routes les plus nouvelles ou autrement dit les plus fraiches (fresh routes). Les nœuds les mis à jour chaque fois qu'une nouvelle information provenant d'un message RREQ, RREP ou RERR [17].

RREQ : contient essentiellement les champs suivants :

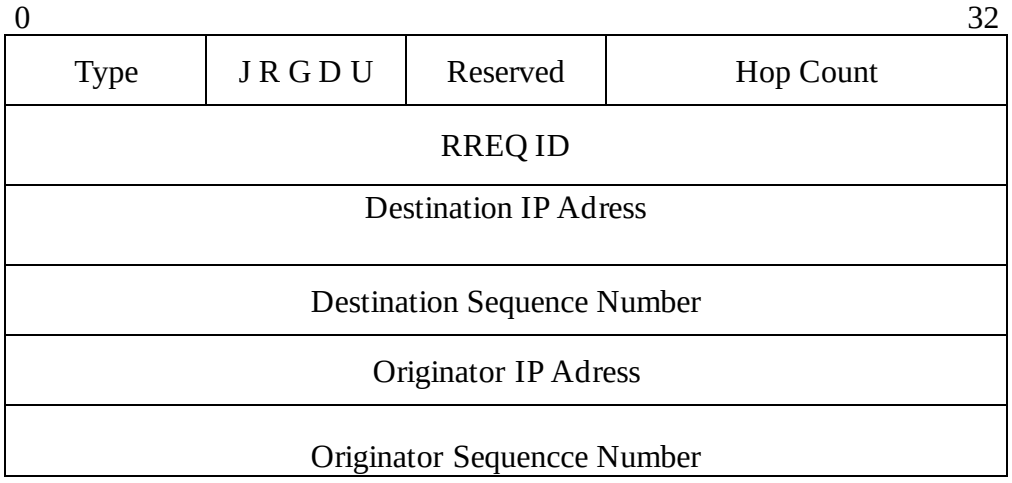


Figure II.1: Format du message Route Request (RREQ)

- **Type (8 bits) :** ce champ indique le type de paquet, dans ce cas il prend la valeur 1
- **Flags (drapeaux) (5 bits) :** ce champ contient cinq flags (J, R, G, D, U).
- **Reserved (11 bits):** initialisé à la valeur 0 et ignoré à la réception du message.
- **Hop Count (8 bits):** il contient le nombre de sauts parcourus par RREQ.
- **RREQ ID:** il identifie la requête parmi les requêtes envoyées par la même source.
- **Destination IP Address :** l'adresse IP de destination pour laquelle une route est désirée.

- **Destination Sequence Number :** Le dernier numéro de séquence reçu dans le passé pour n'importe quelle route vers la destination.
- **Originator IP Adress** : l'adresse IP de la source de la requête.
- **Originator Sequence Number :** Le nombre de séquence courant de la source contenue dans la table de routage de ce noeud.

RREP: contient essentiellement les champs suivants :

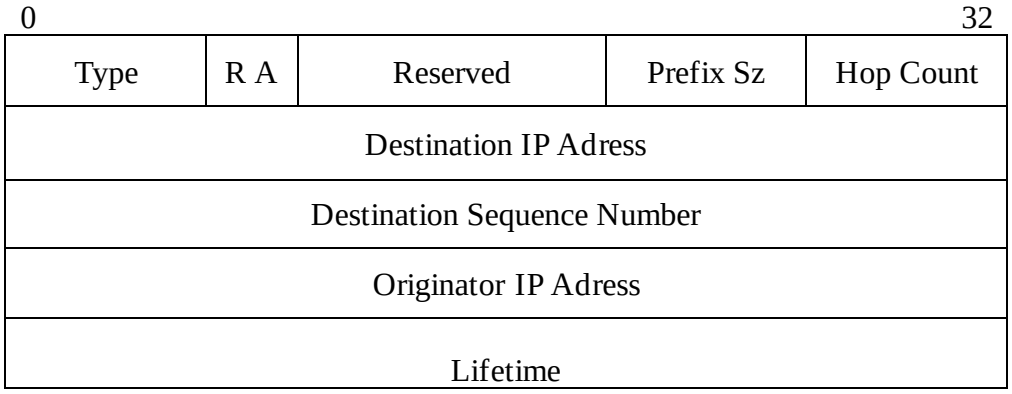


Figure II.2: Format du message Route Reply (RREP)

- **Type (8 bits):** ce champ indique le type de paquet, dans ce cas il prend la valeur 2.
- **Flags :**(drapeaux) **(2 bits)**
- **Reserved (9 bits):** initialisé à la valeur 0 et ignoré à la réception du message.
- **Préfix Sz (5 bits):** si la valeur de ce champs est différente de zéro, ce dernier indique que le nœud prochain peut être utilisé pour chaque nœud demandant cette destination et qui possède la même valeur de Prefix Sz.
- **Hop Count (8 bits):** il contient le nombre de sauts entre la source jusqu'à la destination.
- **Destination IP Address :** l'adresse IP de la destination du paquet RREQ.
- **Destination Sequence Number :** le numéro de séquence de la destination associé à cette route.
- **Originator IP Adress :** l'adresse IP du nœud qui crée la requête.
- **Lifetime :** le temps pour lequel chaque nœud qui reçoit RREP considère que la route est valide.

RERR : Un message d'erreur de route contient essentiellement les champs suivants :

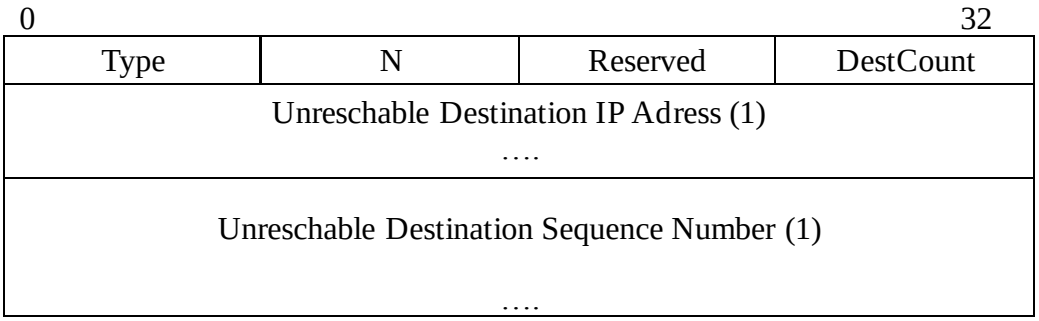


Figure II.3: Route Error (RERR)Format du message.

- **Type (8 bits):** la valeur de ce champ prend 3 dans le message RERR.

- **Flag (1 bits):** il contient un drapeau (N: No delete flag), celui-ci est indicatif lorsqu'un nœud est capable de réparer le lien, et informe les nœuds suivants qu'ils ne doivent pas supprimer le chemin.
- **Reserved (15 bits):** initialisé à la valeur 0 et ignoré à la réception du message.
- **DestCount (8 bits) :** il indique le nombre de destinations inaccessibles incluses dans ce message. Ce champ doit être supérieur ou égal à un.
- **Unreachable Destination IP Address :** l'adresse IP des destinations inaccessibles pour la raison de rupture de lien.
- **Unreachable Destination Sequence Number :** le nombre de séquence de la liste des destinations inaccessibles qui se trouve dans le champ Unreachable Destination IP Address.

II.4.3 Principe de Fonctionnement de protocole :

Lorsqu' un nœud veut émettre un message, il cherche dans sa table de routage si une route valide existe pour la destination qu'il souhaite atteindre, s'il n'en existe aucune, il se met à la recherche d'une route

Quand un nœud source **S** veut atteindre la destination **D** pour laquelle il ne possède pas de route, il inonde le message de demande de route (RREQ) à ses voisins.

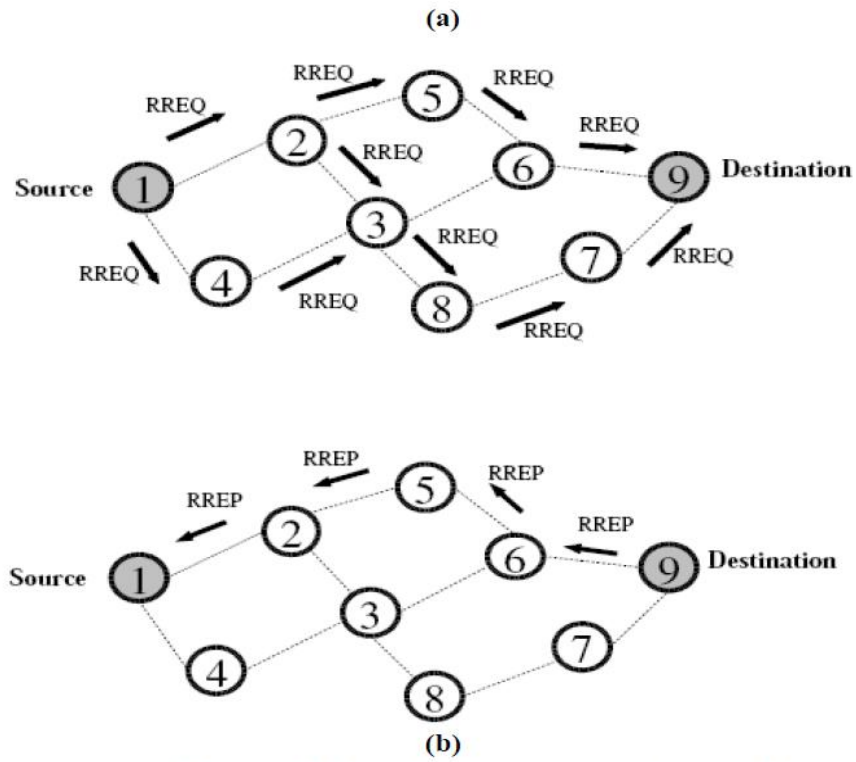
Le paquet RREQ contient le numéro de séquence pour cette destination, si le numéro de séquence n'est pas connu, la valeur nulle sera prise par défaut, il contient aussi la valeur du numéro de séquence du nœud source.

Le RREQ sera propagé jusqu'à ce que le paquet atteigne un nœud qui a une route à la destination ou la destination elle-même.

Chaque nœud intermédiaire expédie la demande à ces voisins et crée une route inversée vers **S** (mémorise une route vers la source).

Quand un nœud intermédiaire a une route vers **D** ou est lui-même la destination **D**, il produit une réponse de route (RREP) qui contient le nombre de saut et le numéro de séquence (le plus récent) pour **D** et le lifetime de la route.

Les nœuds qui portent la réponse vers **S** créent une route vers **D** mais seulement avec le prochain saut et non pas la route toute entière [18]. Cependant, AODV maintient les chemins d'une façon distribuée en gardant une table de routage au niveau de chaque nœud de transit appartenant au chemin recherché.



(a) Inondation de RREQ, (b) revoie du RREP dans AODV.

Figure II.4 : les deux requête RREQ et RREP utilisées dans le protocole AODV

Afin de limiter le coût de la découverte de route dans le réseau, AODV propose d'étendre la recherche progressivement. Initialement, la requête est diffusée à un nombre de sauts limité. Si la source ne reçoit aucune réponse après un délai d'attente déterminé, elle retransmet un autre message de recherche en augmentant le nombre maximum de sauts. En cas de non réponse, cette procédure est répétée un nombre maximum de fois avant de déclarer que cette destination est injoignable.

A chaque nouvelle diffusion, le champ *Broadcast ID* du paquet RREQ est incrémenté pour identifier une requête de route particulière associée à une adresse source. Si la requête RREQ est rediffusée un certain nombre de fois (RREQ_RETRIES) sans la réception de réponse, un message d'erreur est délivré à l'application.

La destination renvoie un message RREP, ce message peut donc être acheminé vers la source. Chaque nœud traversé incrémentera le nombre de sauts. Et ajoutera une entrée à sa table pour la destination.

Une réponse adéquate peut aussi être donnée par un nœud situé entre la source et la destination. Dans ce cas l'obtention de routes bidirectionnelles est néanmoins possible grâce au drapeau "Gratuitous RREP". Le nœud intermédiaire enverra alors en plus un RREP (gratuitos RREP) vers la destination. Les nœuds entre le nœud intermédiaire et la destination ajouteront donc à leur table une entrée vers la source du RREQ. Cette disposition permettra à la destination d'envoyer directement des paquets à la source sans devoir procéder à la recherche d'une route. C'est utile lors de l'établissement de communications TCP pour l'envoi du premier ACK.

II.4.4 Numéro de séquences

Dans AODV, chaque nœud maintient une table qui contient une entrée pour chaque destination accessible. Pour éviter le problème du comptage à l'infini, AODV utilise les numéros de séquences dans les tables de routage en plus de la distance. Chaque nœud possède un numéro de séquence, et il est la seule habilité à l'incrémenter. Ce numéro personnel est incrémenté dans les cas suivants :

- ✓ Avant d'entreprendre un processus de recherche de route, le nœud incrémente son numéro.
- ✓ Avant de répondre à un message RREQ par un message RREP, le numéro de séquence doit être remplacé par la valeur maximale entre son numéro de séquence actuel et celui contenu dans le message RREQ.

Afin de garantit la création de route sans boucles, La mise à jour de la table de routage dans l'AODV ne s'effectue que dans les cas suivants :

- ✓ Le numéro de séquence du paquet de contrôle est strictement supérieur au numéro de séquence présent dans la table.
- ✓ Les numéros de séquence (de la table et du paquet) sont égaux mais, la distance en nombre de sauts du paquet plus 1 est inférieure à la distance actuelle dans la table de routage.
- ✓ Le numéro de séquence pour cette destination est inconnu.

II.4.5 La gestion de la table de routage

L'AODV utilise une requête de route dans le but de créer un chemin vers une certaine destination. Cependant, AODV maintient les chemins d'une façon distribuée, chaque nœud intermédiaire fait partie du chemin recherché, stocke une entrée dans sa propre table de routage.

Chaque entrée de la table de routage contient les informations suivantes :

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

A chaque utilisation d'une entrée, son temps d'expiration (Active Route Lifetime) est mis à jour (temps courant + ACTIVE_ROUTE_TIMEOUT) [19].

II.4.6 Le processus de la découverte de la route par AODV :

Le processus de la découverte de chemin est lancé lorsqu'un nœud désire établir une route vers la destination sur laquelle il ne possède pas encore d'information de sa table de routage.

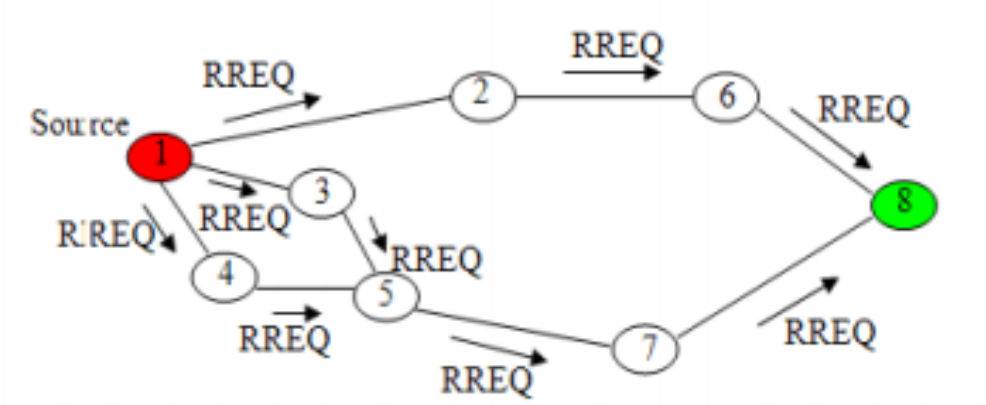
Chaque nœud maintient deux compteurs, « Sequence number » et « Broadcast_id ». ce pair d'informations <Source_addr, Broadcast_ id> est une identification du message RREQ, et le champ broadcast_ id est incrémenté à chaque envoie de message RREQ.

Lorsqu'un nœud reçoit le message RREQ, il émet un paquet route reply RREP s'il est la destination ; sinon s'il possède une route vers la destination avec un numéro de séquence supérieur ou égale à celui indiquer dans RREQ, il transmit (unicast) un paquet RREP vers la source.

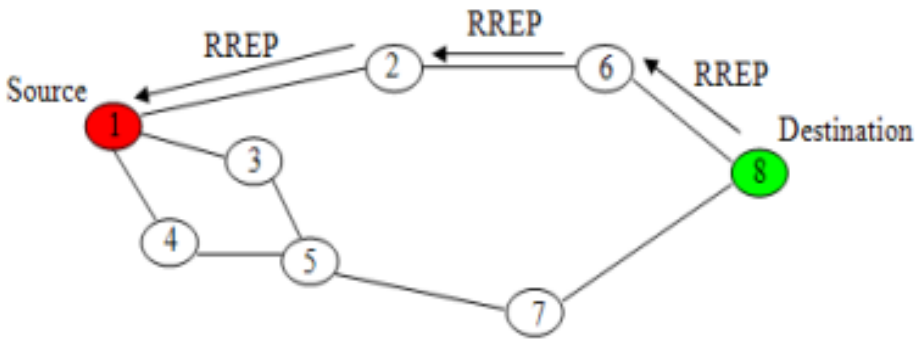
Dans le cas contraire, chaque nœud transmit RREQ à ses propres voisins après avoir incrémenté le compteur de saut « hop_count » en gardant trace de l'adresse IP source, IP destination ainsi le broadcast_id. Si un nœud reçoit un paquet qui a déjà été traité, il le rejette et ne le transmit pas.

Les nœuds établissent des pointeurs de propagation vers la destination alors que les RREP reviennent vers la source. Une fois que la source a reçu RREP, des paquets de données peuvent être émis à la destination.

La figure suivante illustre le processus de la découverte de la route par AODV :



a. La propagation du paquet RREQ (requête de route)



b. La propagation du paquet RREP (requête de réponse)

Figure II.5: Processus de la découverte de la route par AODV

Un nœud diffuse une requête de route (RREQ) pour connaître la route vers une certaine destination si celle-ci n'est pas connue au préalable, ou si le chemin existant vers la destination a expiré sa durée de vie ou il est devenu défaillant.

Le champ numéro de séquence de destination du paquet RREQ, contient la dernière valeur connue du numéro de séquence, recopiée de la table de routage. Si le numéro de séquence n'est pas connu, la valeur nulle sera prise par défaut. Le numéro de séquence source du paquet RREQ contient la valeur du numéro de séquence du nœud source.

Après la diffusion du RREQ, la source attend le paquet réponse de route (RREP). Si ce dernier n'est pas reçu durant une certaine période (appelée RREP_WAIT_TIME), la source peut rediffuser une nouvelle requête RREQ.

Quand un nœud de transit (intermédiaire) envoie le paquet de la requête à un voisin, il sauvegarde aussi l'identificateur du nœud à partir duquel la première copie de la requête est reçue.

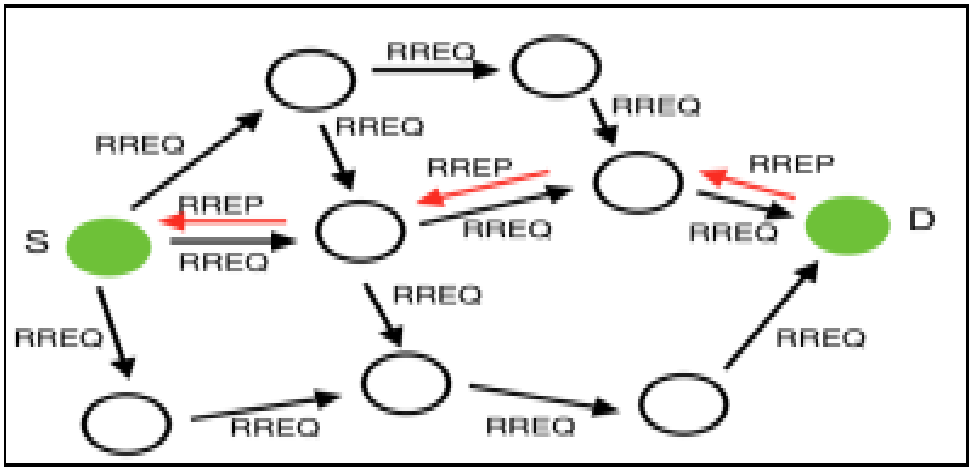


Figure II.6 : Méthode de construction d'une route

Cette information est utilisée pour construire le chemin inverse, qui sera traversé par le paquet réponse de route de manière unicast. Puisque le paquet RREP va être envoyé à la source, les nœuds appartenant au chemin de retour vont modifier leurs tables de routage suivant le chemin contenu dans le paquet de réponse (temps d'expiration, numéro de séquence e prochain saut).

Afin de limiter le coût dans le réseau, AODV propose d'étendre la recherche progressivement. Initialement, la requête est diffusée à un nombre de sauts limité. Si la source ne reçoit aucune réponse après un délai d'attente déterminé, elle retransmet un autre message de recherche en augmentant le nombre maximum de sauts. En cas de non réponse, cette procédure est répétée un nombre maximum de fois avant de déclarer que cette destination est injoignable.

A chaque nouvelle diffusion, le champ Broadcast ID du paquet RREQ est incrémenté pour identifier une requête de route particulière associée à une adresse source. Si la requête RREQ est rediffusée à un certain nombre de fois (RREQ_RETRIES) sans la réception de réponse, un message d'erreur est délivré à l'application.

La destination renvoie un message RREP, ce message peut donc être acheminé vers la source. Chaque nœud traversé incrémentera le nombre de sauts. Et ajoutera une entrée à sa table pour la destination.

Une réponse adéquate peut aussi être donnée par un nœud situé entre la source et la destination. Dans ce cas l'obtention de routes bidirectionnelles est néanmoins possible grâce au drapeau " Gratuitous RREP". Le nœud intermédiaire enverra alors en plus un RREP vers la destination. Les nœuds entre le nœud intermédiaire et la destination ajouteront donc à leur table une entrée vers la source du RREQ. Cette

disposition permettra à la destination d'envoyer directement des paquets à la source sans devoir procéder à la recherche d'une route.

II.4.7 Maintenance des routes :

Lors de la transmission périodique des données de la source vers la destination, la route est considérée active. Une fois la source s'arrête d'émettre des paquets, le lien expirera et il sera effacé des tables de routage des nœuds intermédiaires. Si un lien se rompt alors qu'une route est active, AODV utilise un message HELLO permettant de vérifier la connectivité des routes. Si pendant un laps de temps, trois messages HELLO ne sont pas reçus, alors le lien vers la destination est considéré cassé. Il envoie donc un message d'erreur RERR à la source pour le notifier de la destination désormais inatteignable. Après la réception de RERR, si la source désire toujours la route, il peut ré initier un processus de la découverte de la route [20].

Pour la maintenance des routes, le protocole AODV exige l'échange périodique des messages HELLO toutes les quelques secondes. Un lien est considéré invalide si trois messages HELLO consécutifs ne sont pas reçus (à travers ce même lien) [21].

Quand un lien devient invalide, tout nœud expédiant à travers celui-ci est informé par un paquet Route Error (RERR) avec une métrique égale à l'infini. Ce qui conduit au lancement d'une opération de découverte de route.

Pour chaque route valide maintenue par le nœud sous la forme d'entrée à la table de routage, ce nœud maintient aussi une liste de précurseurs qui peuvent router les paquets sur cette route. Les précurseurs sont ceux qui ont déjà acheminé une réponse RREP vers ce nœud.

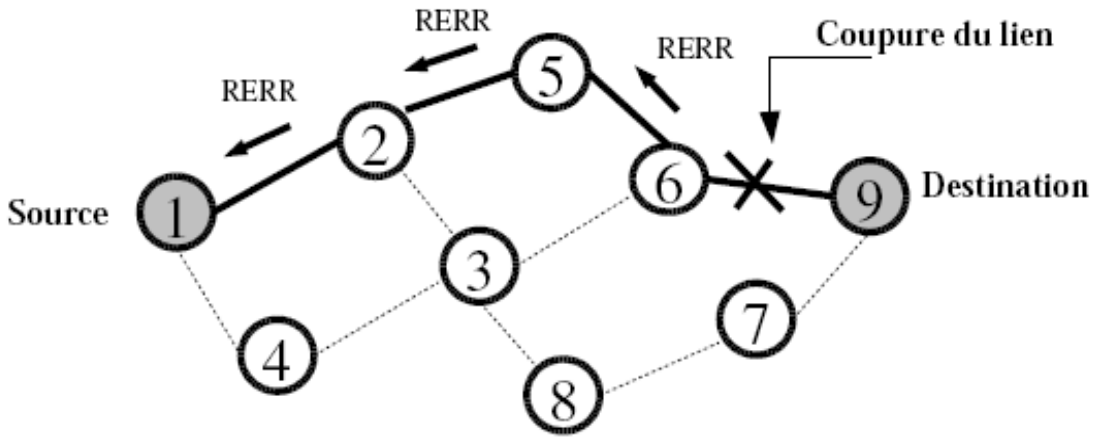


Figure II.7: Un lien devient invalide

II.5 Propriétés d'AODV :

II.5.1 Les avantages d'AODV :

L'un des avantages d'AODV est l'utilisation de numéro de séquence dans les messages. Ces numéros de séquences permettent l'éviter les problèmes de boucles infinies et sont essentiels au processus de mise à jour de la table de routage [22].

Un autre avantage est le rappel de l'adresse IP du nœud origine dans chaque message. Ceci permet de ne pas perdre la trace du nœud à l'origine de l'envoi du message lors des différents relais.

Le protocole de routage AODV n'a pas besoin de système administratif central pour contrôler le processus de routage. Les protocoles réactifs comme AODV ont tendance à réduire le contrôle de la circulation des messages généraux au coût de l'augmentation de la latence à trouver de nouveaux itinéraires (routes).

AODV réagit assez rapidement aux changements topologiques dans le réseau et met à jour uniquement les nœuds affectés par ces changements.

Les messages HELLO assure le maintien de routes en communiquant seulement avec les voisins directs, donc ils ne provoquent pas une surcharge sur le réseau.

Le protocole de routage AODV économise de la mémoire aussi bien que l'énergie. Le nœud de destination répond une seule fois à la première demande et ignore le reste. La table de routage maintient au plus une entrée par destination. Si un nœud doit choisir entre deux trajets, le trajet récent avec un numéro de séquence de destination plus grand, est toujours choisi [23].

Si une entrée de table de routage n'est pas utilisée récemment, l'entrée est expirée après un certain délai et la route devient invalide. La route invalide est supprimée après un délai. : les paquets d'erreur (RERR) atteignent tous les nœuds utilisant le lien coupé situant sur la route vers toute destination.

II.5.2 Les inconvénients d'AODV :

Un inconvénient d'AODV est qu'il n'existe pas de format générique des messages. Chaque message a son propre format : RREQ, RREP, RERR, HELLO, gratuitos RREP [22].

Il est possible qu'une route valable soit expirée. La détermination d'un temps d'expiration raisonnable est difficile, parce que les nœuds sont mobiles ce qui nécessite de relancer le processus de découverte de route et l'envoi des RREQ avec un taux différent selon la mobilité du réseau.

En outre, AODV recueille une quantité très limitée d'informations de routage, l'acquisition des informations de routage sont obtenue uniquement des paquets de contrôles. Ceci provoque une inondation de découverte de route plus fréquemment, ce qui peut entraîner d'importantes surcharges réseau. Les inondations incontrôlées génèrent de nombreuses transmissions redondantes qui peuvent provoquer ce qu'on appelle « broadcast storm »

La performance du protocole AODV se conduisant mal dans de plus grands réseaux. La principale différence entre les petits et les grands réseaux est la longueur moyenne des chemins. Un long chemin est plus vulnérable aux ruptures de lien et requiert une charge de contrôle élevé pour son entretien [23].

En outre, comme la taille d'un réseau grandit, diverses mesures de performance commencent à diminuer en raison de l'augmentation du travail administratif, que l'on appelle la charge administrative.

AODV est vulnérable à toutes sortes d'attaques, car il repose sur l'hypothèse que tous les nœuds vont coopérer. Sans cette coopération, aucune route ne peut être établie et aucun paquet ne peut être transmis. Il existe deux principaux types de nœuds non coopératifs : malveillants et égoïstes. Les nœuds malveillants sont soit défectueux et ne peuvent pas suivre le protocole, ou sont intentionnellement malveillant et essaient d'attaquer le réseau.

L'égoïsme est la non coopération dans de certaines opérations de réseau, par exemple : suppression de paquets qui peuvent affecter les performances, mais peuvent économiser la batterie.

II.6 Vulnérabilités dans AODV :

Le protocole AODV est très efficace en tant que service de réseau, mais il a beaucoup de vulnérabilités, signifie que ce protocole peut facilement être attaqué. AODV n'est pas si sécurisé. AODV est conçu pour un réseau idéal signifie pour un réseau n'ayant aucun nœud malveillant. Pour un réseau n'ayant aucun nœud malveillant le protocole AODV est le plus efficace. Mais nous savons tous que rien n'est idéal, autrement dit qu'il y a certains nœuds incommodes partout. Quelques nœuds gourmands peuvent exister aussi ce qui met en échec l'objectif de l'utilisation du réseau.

Afin de lancer une attaque contre le protocole de routage AODV, il suffit de faire des modifications dans les messages RREQ, RREP, RERR comme suit [24] :

- Les numéros de séquence peuvent être modifiés.
- Le nombre de sauts peuvent être modifié. (Principale attaque est la formation des boucles dans le réseau « Looping »).
- Modification des routes source (attaque Blackhole, des informations erronées sur le chemin).
- Tunneling (warm hole).
- Spoofing (collecte d'information sur le routage).

II.7 Les différentes attaques de routage AODV :

II.7.1 Le modèle d'un attaquant :

La première étape pour sécuriser un système est l'identification de la nature des éventuels attaquants. Dans les réseaux ad hoc, nous pouvons classifier un attaquant selon les dimensions suivantes **:**

- **Interne vs. Externe :**

L'attaquant interne est perçu comme un membre normal du réseau et peut communiquer avec les autres membres. La présence des attaques internes est très problématique et difficile à détecter, car elle annule le niveau de sécurité assuré par les techniques cryptographiques. L'attaquant externe est considéré par les nœuds membres comme un intrus et est donc limité dans la diversité des attaques qu'il peut provoquer[25].

- **Malveillant vs Rationnel :**

Un attaquant malveillant n'a pas d'intérêts personnels à travers ses attaques et a pour but le dysfonctionnement du réseau. Par conséquent, il peut employer tous les moyens sans tenir compte des coûts correspondants et des conséquences. Par contre, un attaquant rationnel cherche un profit personnel, et ainsi, on peut prévoir les cibles d'attaques et les moyens employés**.**

- **Passif vs. Actif :**

L'attaquant passif écoute simplement les informations qui sont échangées entre les nœuds tandis que l'attaquant actif agit sur les informations qui sont échangées. Il peut les falsifier, les modifier, voire même les détruire.

II.7.2 Attaque de largage de paquets :

Dans une attaque d'abandon de paquet, les messages de routage reçus sont simplement abandonnés par l'attaquant. Cela peut être détecté en surveillant si un nœud voisin diffuse des paquets vers la destination finale ou non. Pour activer la surveillance des nœuds voisins, il est nécessaire de conserver la table des voisins.

Cette attaque est disponible sous différentes formes. Les différentes sous-catégories sont les suivantes **:**

Si un attaquant veut appliquer une attaque de suppression de paquets sur les messages RREQ qu'il reçoit, les messages RREQ peuvent également être supprimés de manière sélective par un attaquant interne. De tels types d'abus par des attaquants sont de nature similaire aux nœuds égoïstes. Si un attaquant s'inquiète de l'application de cette attaque sur les messages RREP, cela peut être le cas d'une perturbation d'itinéraire. Cette attaque peut également s'appliquer à d'autres paquets de données et empêchera le nœud affecté de prendre des paquets de données des nœuds voisins pendant une courte période de temps. Après avoir reçu le message RREQ, un attaquant peut apporter des modifications comme augmenter l'ID RREQ, changer l'adresse IP de destination avec une autre adresse IP, ajouter le numéro de séquence source par un, mettre une adresse IP inexistante à la place de l'adresse IP source. Après cela, un faux message peut être transmis par un attaquant.

Lorsque tous les voisins d'un attaquant reçoivent le faux message RREQ, ils modifient le saut suivant du nœud source vers le nœud inexistant car le faux message RREQ a un numéro de séquence source plus grand. En raison d'une adresse IP de destination inexistante, le faux message se rendra aux nœuds extrêmes du réseau ad hoc. Chaque fois qu'un nœud nécessite l'envoi de paquets de données vers le nœud source, il suivra la route créée à l'aide du faux message RREQ. En raison d'un nœud inexistant, les paquets de données peuvent être abandonnés. Avec l'aide de mécanismes de réparation locaux dans le protocole AODV, cette attaque ne peut pas totalement séparer le nœud victime. Chaque fois qu'un nœud observe une livraison infructueuse de paquets de données, les nœuds recommencent le processus de découverte d'itinéraire [26]**.**

II.7.3 Attaque par numéro de séquence :

La fraîcheur de la route couplée à un nœud sera indiquée en utilisant le numéro de séquence. Si un attaquant transmet un paquet de contrôle AODV avec un grand numéro de séquence du nœud compromis, la route sera dirigée vers le nœud compromis. Le numéro de séquence peut être réduit pour restreindre la mise à jour dans la table ou augmenté pour renouveler les tables de route inverse d'autres nœuds. Cette attaque peut également s'appliquer à la fois au numéro de séquence source ou au numéro de séquence de destination. Un message RREQ peut être identifié de manière unique par l'ID RREQ avec l'adresse IP source. La combinaison de cela montre la fraîcheur d'un message RREQ. À tout moment, un nœud ne considère que la première copie d'un message RREQ. Si un autre nœud accepte l'ID RREQ augmenté avec l'adresse IP source, cela signifie que le nœud acceptera le faux message RREQ. L'attaque par numéro de séquence ne peut prendre en compte que le champ de numéro de séquence, disponible dans le message RREQ[26].

II.7.4 Attaque de modification de champ :

Comme on le sait, le paquet de données sera envoyé avec l'en-tête. Dans le processus de superposition, les paquets de données passent par différentes couches et ajoutent des en-têtes en conséquence. L'attaque

de modification de champ est responsable de la modification des valeurs de champ dans l'en-tête au niveau de la couche réseau. Comme ci-dessus, l'attaque par numéro de séquence modifie le champ de numéro de séquence, par conséquent, on peut dire que l'attaque par numéro de séquence fait partie de l'attaque de modification de champ. Les autres champs qu'un attaquant peut modifier sont mis en évidence ci-dessous. Le tableau ci- dessous montrera l'impact du champ modifié pendant le processus de routage normal.

Champ de message RREQ	Modifications
RREQ ID	Pour rendre acceptable ou inacceptable un faux message RREQ, l'attaquant augmente ou diminue l'ID RREQ.
Type	Le type de message sera modifié.
Hop Count	Pour invalider la mise à jour, le nombre de sauts sera diminué ou augmenté pour mettre à jour les tables de routage inversé des autres nœuds.
Destination IP Address	Remplacer par une autre adresse IP
Source IP Address	Remplacez par une autre adresse IP pour modifier l'itinéraire inverse

Tableau II.1 : Attaque de modification de champ sur le champ de message RREQ

Lorsque plusieurs champs ont été modifiés par l'attaquant, cela montre des répercussions immédiates sur la sécurité du réseau. Pour garantir l'absence de boucle dans AODV, un nœud après avoir reçu un message RREQ modifie sa table de routage inverse. Cette modification se produit uniquement si le numéro de séquence source est supérieur à la valeur dans sa table de routage ou si le numéro de séquence source est égal mais la valeur du nombre de sauts est inférieure à celle de la table de routage pour le message RREQ.

Pour affecter la table de routage des autres nœuds, un attaquant interne peut également impliquer la modification de ces champs.

La même procédure sera utilisée pour un message RREP. Dans ce cas, si le numéro de séquence de destination dans le message RREP est supérieur à la valeur un dans sa table de routage ou le numéro de séquence de destination est le même mais que le nombre de sauts plus un est inférieur à la valeur dans la table de routage, un nœud source ou un nœud intermédiaire modifie sa table de routage avant. Maintenant, prenez le point de vue de l'attaquant, si le numéro de séquence de destination dans le message RREP est supérieur à celui de sa table de routage, ou si les numéros de séquence de destination sont les mêmes, mais le nombre de sauts dans le message RREP plus un est plus petit que celui dans sa table de routage, l'attaquant peut contenir le message RREP légitime en augmentant le numéro de séquence de destination [26].

II.7.5 Attaque d'ajout de champ :

Dans cette attaque, un attaquant peut construire un message RREQ sans recevoir de message RREQ. Pour lancer cette attaque, il est nécessaire de collecter des informations de base pour créer de faux messages RREQ (par exemple, en écoutant le trafic). En théorie, pour perturber le processus de routage, l'attaquant peut ajouter n'importe quel champ dans un message RREQ [26].

II.8 Conclusion :

Dans ce chapitre nous avons étudié en détail le fonctionnement et le comportement de protocole de routage AODV dans les réseaux ad hoc et les différentes attaques possibles contre ce protocole de routage.

En effet, on peut conclure que le protocole de routage AODV n'assure pas totalement la sécurité de routage contre les différentes attaques.

Donc, d'autres mécanismes doivent être mis en œuvre afin d'améliorer sécurité de routage et la disponibilité de réseau.

Dans le chapitre qui suit nous allons présenter le protocole IPsec (Internet Protocol Security) et son fonctionnement, IPsec est un protocole qui permet de sécuriser les échanges sur un réseau IP.

Chapitre III : La sécurité de protocole avec IPsec

III.1 Introduction :

Pour sécuriser les échanges sur un réseau Internet, Il existe plusieurs solutions de sécurité pour sécuriser les communications Internet, notamment l'IPSec (IP Security), le SSL/TLS (Socket Secure Layer/Transport Layer Security) et le SSH (Secure Shell)[27]. En fait, il ne s'agit plus seulement de rechercher de nouveaux développements pour rendre le réseau plus sûr ou, en général, plus sûr dans son fonctionnement global. Ces solutions doivent être adaptées aux besoins spécifiques des utilisateurs ainsi qu'à leurs environnements.

Dans ce chapitre nous présentons le protocole IPsec (Internet Protocol Security), son fonctionnement et les services de sécurité fournis par ce protocole.

III.2 Définition d'IPsec :

IPsec (Internet Protocol Security), défini par l'IETF comme un cadre de standards ouverts pour assurer des communications privées et protégées sur des réseaux IP, par l'utilisation des services de sécurité cryptographiques, est une suite de protocoles destinée à permettre une communication sécurisée sur des réseaux IP potentiellement non sécurisés tels qu'Internet.

IPSec fonctionne directement sur la couche réseau (couche Internet) du modèle DoD (Department of Defense) et constitue un développement ultérieur des protocoles IP. L'objectif est de fournir une sécurité basée sur le chiffrement au niveau du réseau. IPSec offre cette possibilité grâce à l'intégrité en mode sans connexion ainsi qu'au contrôle d'accès et à l'authentification des données. De plus, IPsec garantit la confidentialité et l'authenticité de la séquence de paquets grâce au cryptage [28].

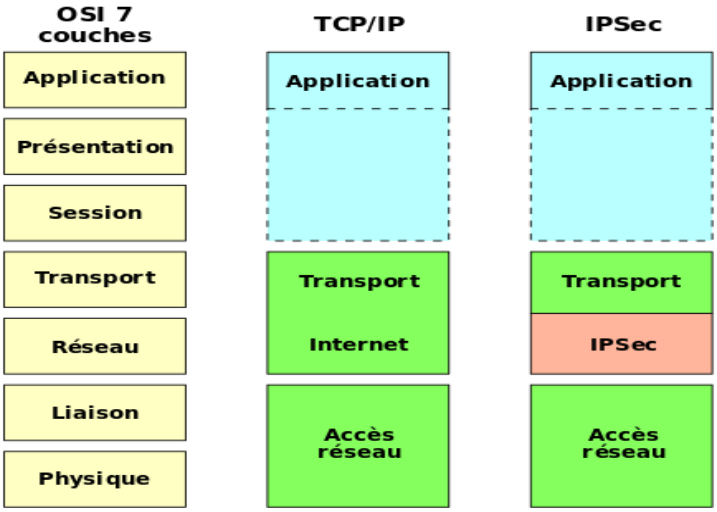


Figure III.1: Positionnement protocole IPsec dans le modèle OSI

III.3 Services de sécurité fournis par IPsec :

Le protocole IPSec est destiné à fournir différents services de sécurité. Il permet grâce à plusieurs choix et options de définir différents niveaux de sécurité afin de répondre de façon adaptée aux besoins de chaque entreprise.

La stratégie IPSec permettant d'assurer la confidentialité, l'intégrité et l'authentification des données entre deux hôtes est gérée par un ensemble de normes et de protocoles [29]:

- **Confidentialité** :

Le principe de la confidentialité assure que seuls les nœuds authentifiés communicants sont en mesure de comprendre les données secrètes échangées. Des méthodes de contrôles d'accès stricts doivent être mise en place pour assurer la confidentialité des données échangées au sein des nœuds réseaux. La capture des paquets ne doit pas permettre de savoir quelles sont les informations échangées. Seules les machines réceptrices et émettrices doivent pouvoir accéder à l'information.

- **Authentification** :

L'identité des nœuds doit être vérifiable ce qui interdit aux nœuds malveillants non authentifiés d'injecter des messages falsifiés (s'assurer qu'il n'y a pas de nœud intrus qui est masqué usurpant l'identité d'un autre). Le récepteur doit être capable de vérifier si les données reçues proviennent bien de l'émetteur supposé.

- **Intégrité** :

L'intégrité des données garantit que les données échangées n'ont pas été altérées ou modifiées ou détruites durant la communication d'une façon volontaire ou accidentelle. Le récepteur doit être capable de vérifier si les données n'ont pas été modifiées lors de la transmission.

- **Protection contre le rejeu** :

Une personne qui intercepte un message d'une communication sécurisée entre deux machines ne pourra pas retransmettre ce message sans que cela soit détecté.

- **Gestion des clés :**

Mécanisme de négociation de la longueur des clés de chiffrement entre deux éléments IPSEC et d'échange de ces clés.

- **La non-répudiation :**

La non-répudiation assure qu'un message envoyé ne sera pas niée par son expéditeur, cette propriété est réalisée en appliquant une méthode basée sur la signature électronique. Cet objectif de sécurité permet de s'assurer qu'aucun émetteur ne peut nier d'être à l'origine d'un message. Cet objectif est indispensable dans les transactions électroniques et dans toutes les communications sensibles.

III.4 Architecture d'IPsec :

Pour sécuriser les échanges ayant lieu sur un réseau TCP/IP, il existe plusieurs approches, en particulier en ce qui concerne le niveau auquel est effectuée la sécurisation : Niveau applicatif (mails chiffrés par exemple), niveau transport (TLS/SSL, SSH…), ou à l'opposé niveau physique (boîtiers chiffrant toutes les données transitant par un lien donné). IPsec, quant à lui, vise à sécuriser les échanges au niveau de la couche réseau (**figure III .2**).

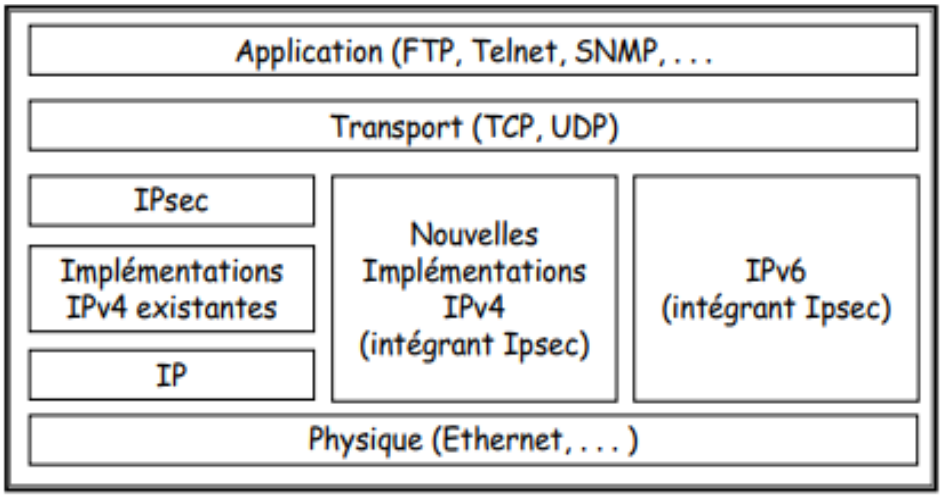


Figure III.2: L'architecteur de protocole IPsec

III.5 La notion d'association de sécurité :

Les mécanismes mentionnés ci-dessus, bien sûr, utilisent la cryptographie, et en conséquence, ils utilisent un ensemble de paramètres (algorithmes cryptographiques, chiffrements, mécanismes, etc.) sur lesquels les parties communicantes doivent s'entendre. IPsec utilise le concept d'association de sécurité pour gérer ces paramètres (Security Association, SA).

Une association de sécurité IPsec est une "connexion" simplexe qui fournit des services de sécurité au trafic qu'elle transporte. On peut aussi la considérer comme une structure de données servant à stocker l'ensemble des paramètres associés à une communication donnée.

Une SA est unidirectionnelle ; en conséquence, protéger les deux sens d'une communication classique requiert deux associations, une dans chaque sens. Les services de sécurité sont fournis par l'utilisation soit de AH soit de ESP. Si AH et ESP sont tous deux appliqués au trafic en question, deux SA sont créées ; on parle alors de paquet (bundle) de SA.

Chaque association est identifiée de manière unique à l'aide d'un triplet composé de :

- L'adresse de destination des paquets.
- L'identifiant d'un protocole de sécurité utilisé (AH ou ESP).
- Un index des paramètres de sécurité (Security Parameter Index, SPI).

Un SPI est un bloc de 32 bits inscrit en clair dans l'en-tête de chaque paquet échangé ; il est choisi par le récepteur.

Pour gérer les associations de sécurité actives, on utilise une "base de données des associations de sécurité" (Security Association Database, SAD). Elle contient tous les paramètres relatifs à chaque SA et sera consultée pour savoir comment traiter chaque paquet reçu ou à émettre.

III.5.1 La gestion des clefs et des associations de sécurité :

Comme nous l'avons mentionné au paragraphe précédent, les SA contiennent tous les paramètres nécessaires à IPsec, notamment les clefs utilisées. La gestion des clefs pour IPsec n'est liée aux autres mécanismes de sécurité de IPsec que par le biais des SA.

Une SA peut être configurée manuellement dans le cas d'une situation simple, mais la règle générale est d'utiliser un protocole spécifique qui permet la négociation dynamique des SA et notamment l'échange des clefs de session. D'autre part, IPv6 n'est pas destiné à supporter une gestion des clefs "en bande", c'est-à-dire où les données relatives à la gestion des clefs seraient transportées à l'aide d'un en-tête IPv6 distinct. Au lieu de cela on utilise un système de gestion des clefs dit "hors bande", où les données relatives à la gestion des clefs sont transportées par un protocole de couche supérieure tel qu'UDP ou TCP.

Ceci permet le découplage clair du mécanisme de gestion des clefs et des autres mécanismes de sécurité. Il est ainsi possible de substituer une méthode de gestion des clefs à une autre sans avoir à modifier les implémentations des autres mécanismes de sécurité.

Le protocole de négociation des SA développé pour IPsec s'appelle "protocole de gestion des clefs et des associations de sécurité Internet" ISAKMP et est en fait inutilisable seul : C'est un cadre générique qui permet l'utilisation de plusieurs protocoles d'échange de clef et qui peut être utilisé pour d'autres mécanismes de sécurité que ceux de IPsec.

Dans le cadre de la standardisation d'IPsec, ISAKMP est associé à une partie des protocoles SKEME et Oakley pour donner un protocole final du nom d'IKE (Internet Key Exchange).

III.5.2 Principe de fonctionnement :

La **(figure III.3)**, ci-dessous, représente tous les éléments présentés ci-dessus (en bleu), leurs positions et leurs interactions.

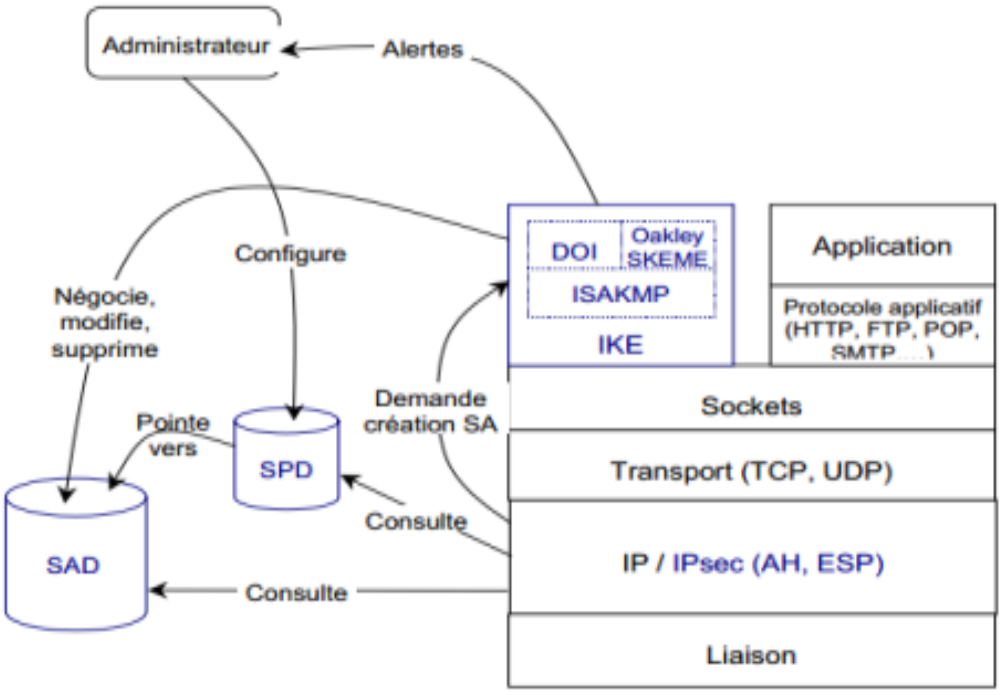


Figure III.3: Schéma global d'IPsec

On distingue deux situations [30] :

- **Trafic sortant:**

Lorsque la "couche" IPsec reçoit des données à envoyer, elle commence par consulter la base de données des politiques de sécurité (SPD) pour savoir comment traiter ces données. Si cette base lui indique que le trafic doit se voir appliquer des mécanismes de sécurité, elle récupère les caractéristiques requises pour la SA correspondante et va consulter la base des SA (SAD). Si la SA nécessaire existe déjà, elle est utilisée pour traiter le trafic en question. Dans le cas contraire, IPsec fait appel à IKE pour établir une nouvelle SA avec les caractéristiques requises.

- **Trafic entrant** :

Lorsque la couche IPsec reçoit un paquet en provenance du réseau, elle examine l'en-tête pour savoir si ce paquet s'est vu appliquer un ou plusieurs services d'IPsec. La cas échéant quelles sont les références de la SA. Elle consulte alors la SAD pour connaître les paramètres à utiliser pour la vérification et/ou le déchiffrement du paquet. Une fois le paquet vérifié et/ou déchiffré, la SPD est consultée pour savoir si la SA appliquée au paquet correspondait bien à celle requise par les politiques de sécurité. Dans le cas où le paquet reçu est un paquet IP classique, la SPD permet de savoir s'il a néanmoins le droit de passer. Par exemple, les paquets IKE sont une exception. Ils sont traités par IKE, qui peut envoyer des alertes administratives en cas de tentative de connexion infructueuse.

III.6 Les Modes de fonctionnement d'IPSec :

Les normes IPsec définissent trois modes distincts d'opération IPsec :

- ✓ Le mode Transport
- ✓ Le mode Tunnel
- ✓ Mode Nesting

- **Le mode transport :**

Dans le mode transport, IPSec intervient entre le niveau transport (TCP) et le niveau réseau (IP) du modèle OSI : le PDU de la couche transport se voit appliqué les mécanismes de signature et de chiffrement puis le résultat est passé à la couche réseau (encapsulation IP). Ce mode ne résout pas un problème majeur en matière de sécurité : l'en-tête du paquet est inchangé puisque produit par la couche IP. Il n'y a donc ni de masquage d'adresse ni de protection des options IP. Cependant ce mode est relativement aisé à mettre en œuvre. En mode transport, seules les données en provenance du protocole de niveau supérieur et transportées par le datagramme IP sont protégées. Ce mode n'est utilisable que sur des équipements terminaux ; en effet, en cas d'utilisation sur des équipements intermédiaires, on courrait le risque, suivant les aléas du routage, que le paquet atteigne sa destination finale sans avoir traversé la passerelle sensée le déchiffrer [31].

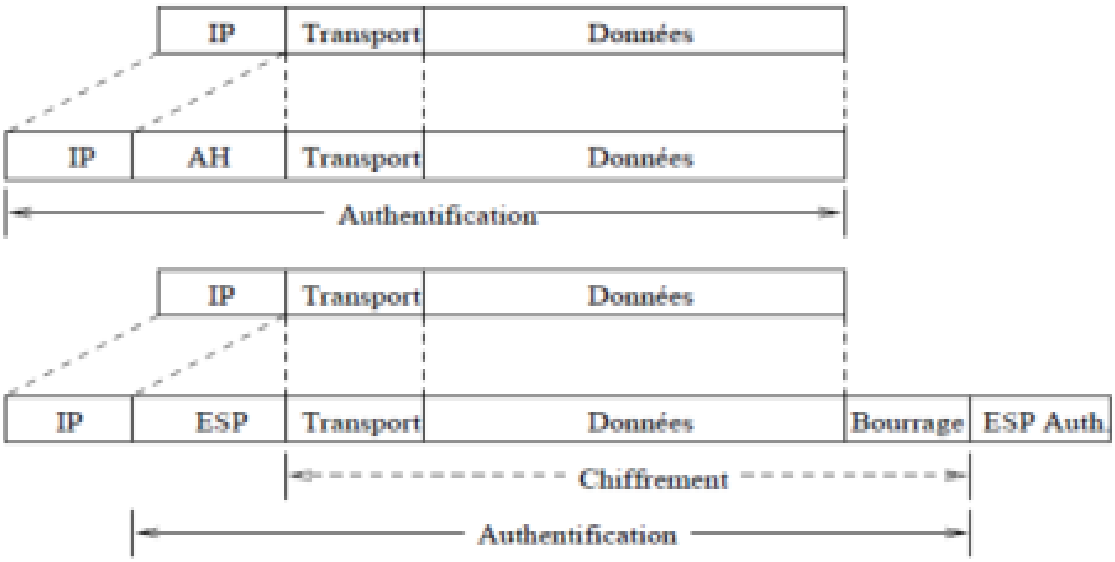


Figure III.4: Paquet IP en mode transport

- **Le mode tunnel :**

Dans le mode tunnel, IPSec agit directement après l'encapsulation IP. La totalité du paquet IP est encapsulé dans un paquet IPSec sécurisé. Dans ce cas, l'en-tête IP d'origine est protégé et les adresses sont masquées. Ce mode est très utilisé pour la mise en place de VPNs.

En mode tunnel, l'en-tête IP est également protégé (authentification, intégrité et/ou confidentialité) et remplacé par un nouvel en-tête. Ce nouvel en-tête sert à transporter le paquet jusqu'à la fin du tunnel, où l'en-tête original est rétabli. Le mode tunnel est donc utilisable à la fois sur des équipements terminaux et sur des passerelles de sécurité. Ce mode permet d'assurer une protection plus importante contre l'analyse du trafic, car il masque les adresses de l'expéditeur et du destinataire final [31].

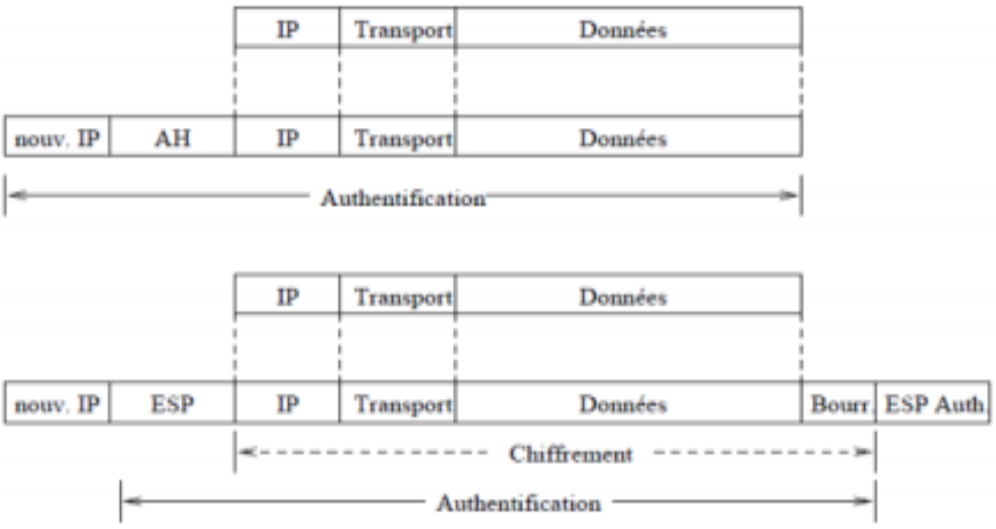


Figure III.5: Paquet IP en mode tunnel

- **Mode Nesting :**

Le mode de Nesting utilise à la fois le mode transport et le mode tunnel : Un paquet IPSec et encapsulé dans un paquet IPSec.

III.7 Les protocoles à la base d'IPsec :

IPsec fait appel à deux mécanismes de sécurité pour le trafic IP, les "protocoles" AH et ESP, qui viennent s'ajouter au traitement IP classique :

- **Authentication Header (AH) :** est conçu pour assurer l'intégrité et l'authentification des datagrammes IP sans chiffrement des données (i.e. sans confidentialité). Le principe d'AH est d'adjoindre au datagramme IP classique un champ supplémentaire permettant à la réception de vérifier l'authenticité des données incluses dans le datagramme.

- **Encapsulating Security Payload (ESP) :** a pour rôle premier d'assurer la confidentialité, mais peut aussi assurer l'authenticité des données. Le principe d'ESP est de générer, à partir d'un datagramme IP classique, un nouveau datagramme dans lequel les données et éventuellement l'en-tête original sont chiffrés.

III.7.1 Les mécanismes de sécurité AH et ESP :

III.7.1.1 Le mécanisme AH :

AH assure l'intégrité des données en mode non connecté, l'authentification de l'origine des données et, de façon optionnelle, la protection contre le rejeu. L'absence de confidentialité dans AH permet de s'assurer que ce standard pourra être largement répandu sur Internet, y compris dans les endroits où l'exportation, l'importation ou l'utilisation du chiffrement dans des buts de confidentialité est restreint par la loi. Cela constitue l'une des raisons de l'utilisation de deux mécanismes distincts.

Dans **AH**, intégrité et authentification sont fournies ensembles, à l'aide d'un bloc de données supplémentaire adjoint au message à protéger. Ce bloc de données est appelé "valeur de vérification d'intégrité (Integrity Check Value, ICV), terme générique pour désigner soit un code d'authentification de message (Message Authentication Code, MAC), soit une signature numérique.

Pour des raisons de performances, les algorithmes proposés actuellement sont tous des algorithmes de scellement (code d'authentification de message et non signature).

La protection contre le rejeu se fait grâce à un numéro de séquence ; elle n'est disponible que si IKE est utilisé, car en mode manuel aucune "ouverture de connexion" ne permet d'initialiser le compteur. Voici l'organisation de l'en-tête d'authentification :

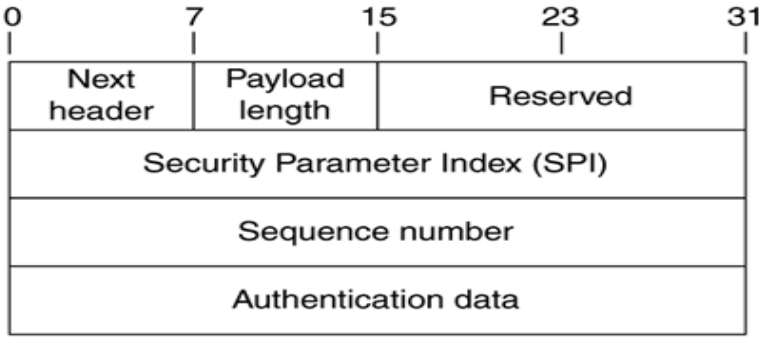


Figure III.6 : La forme En-tête d'un AH

- **Entête suivante :** ce champ permet de spécifier le type du protocole supérieur.
- **Longueur des données :** longueur de l'entête AH par facteur de 32-bit, le minimum étant 2.
- **SPI :** index unique définissant la SA pour ce paquet.
- **Numéros de séquence :** compteur utile au mécanisme d'anti-répétition.
- **Données Authentification (variable) :** champs contenant les signatures de hachages permettant d'authentifier l'émetteur et l'authenticité des données. La taille de ce champ dépend des algorithmes de hachage utilisés.

L'expéditeur calcule les données d'authentification à partir de l'ensemble des champs invariants du datagramme IP final, AH compris, ce qui permet d'étendre l'authentification au SPI et au numéro de séquence notamment [32].

Les champs variables (TTL, routage...) et le champ destiné à recevoir les données d'authentification sont considérés comme égaux à zéro pour le calcul. Les données d'authentification sont alors adjointes au paquet IP par le biais de l'en-tête d'authentification (**AH**). Le récepteur vérifie l'exactitude de ces données à la réception. Les paramètres de sécurité liés à la communication sont identifiés par un identifiant unique (Security Parameters Index) caractérisant la Security Association (SA).

Les figures ci-dessous indiquent la position de **AH** et le service apporté en fonction du mode choisi (transport ou tunnel)

- **Transport Mode (AH) :**

Le mode de transport est simple, il ajoute simplement un en-tête **AH** après l'en-tête IP. Voici un exemple de paquet IP qui transporte du trafic TCP :

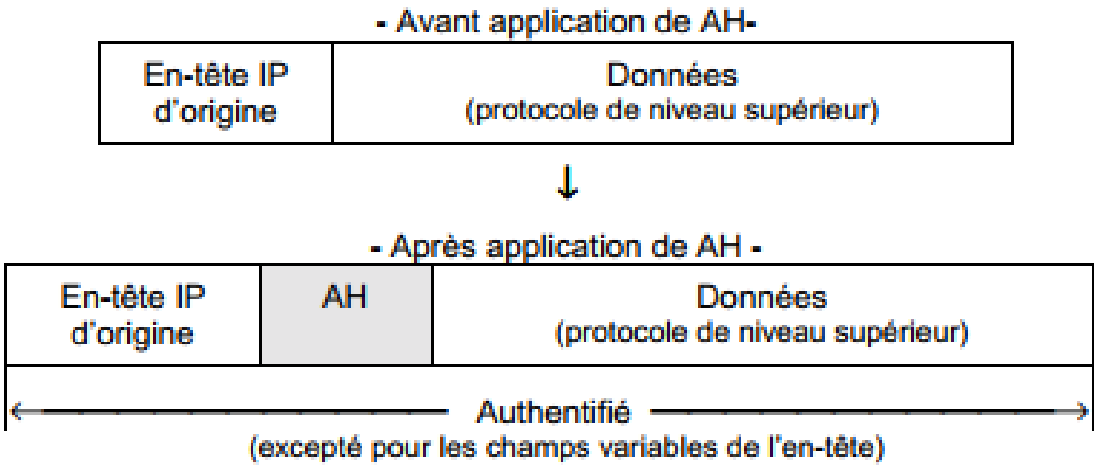


Figure III.7: position de AH en mode transport (ipv4)

- **Tunnel Mode (AH) :**

Avec le mode tunnel, nous ajoutons un nouvel en-tête IP au-dessus du paquet IP d'origine. Cela peut être utile lorsque vous utilisez des adresses IP privées et que vous devez canaliser votre trafic sur Internet. C'est possible avec **AH**, mais il n'offre pas de chiffrement :

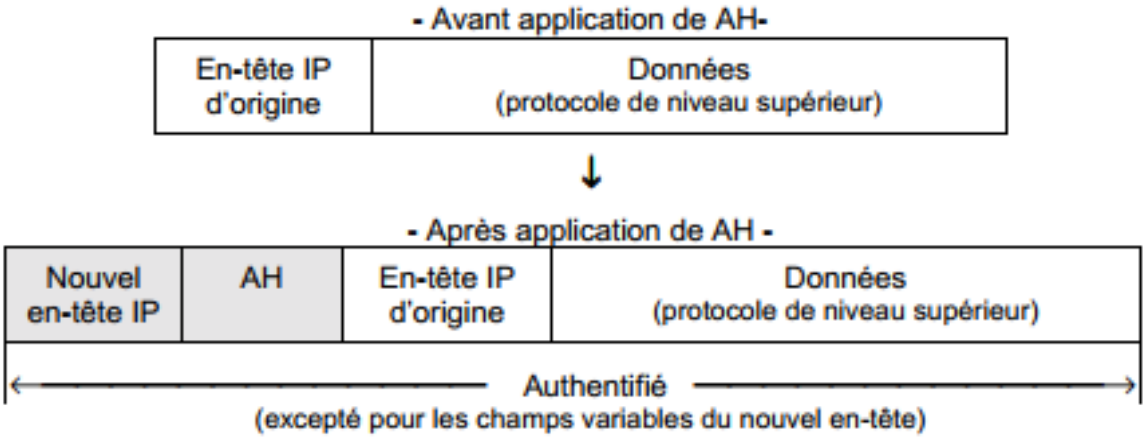


Figure III.8: la position de AH en mode tunnel (ipv4)

Les algorithmes par défaut que doit fournir tout réalisation d'IPSec pour **AH** sont, au moment, HMAC-MD5 et HMAC-SHA-1. Les autres algorithmes prévus sont KPDK-MD5, DES-MAC, HMAC-MD.

III.7.1.2 Le mécanisme (ESP) :

ESP peut assurer, au choix, un ou plusieurs des services suivants :

- Confidentialité (confidentialité des données et protection partielle contre l'analyse du trafic si l'on utilise le mode tunnel).
- Intégrité des données en mode non connecté et authentification de l'origine des données, protection contre le rejeu.

La confidentialité peut être sélectionnée indépendamment des autres services, mais son utilisation sans intégrité/authentification (directement dans **ESP** ou avec **AH**) rend le trafic vulnérable à certains types d'attaques actives qui pourraient affaiblir le service de confidentialité.

Comme dans **AH**, authentification et intégrité sont deux services qui vont de pair et que l'on désigne souvent sous le terme "authentification" ; ils sont fournis par l'utilisation d'une ICV (en pratique, un MAC). La protection contre le rejeu ne peut être sélectionnée que si l'authentification l'a été et que IKE est utilisé. Elle est fournie par un numéro de séquence que le destinataire des paquets vérifie.

Contrairement à **AH**, où l'on se contentait d'ajouter un en-tête supplémentaire au paquet IP, **ESP** fonctionne suivant le principe de l'encapsulation : les données originales sont chiffrées puis encapsulées entre un en-tête et un trailer.

Voici l'organisation d'ESP [33] :

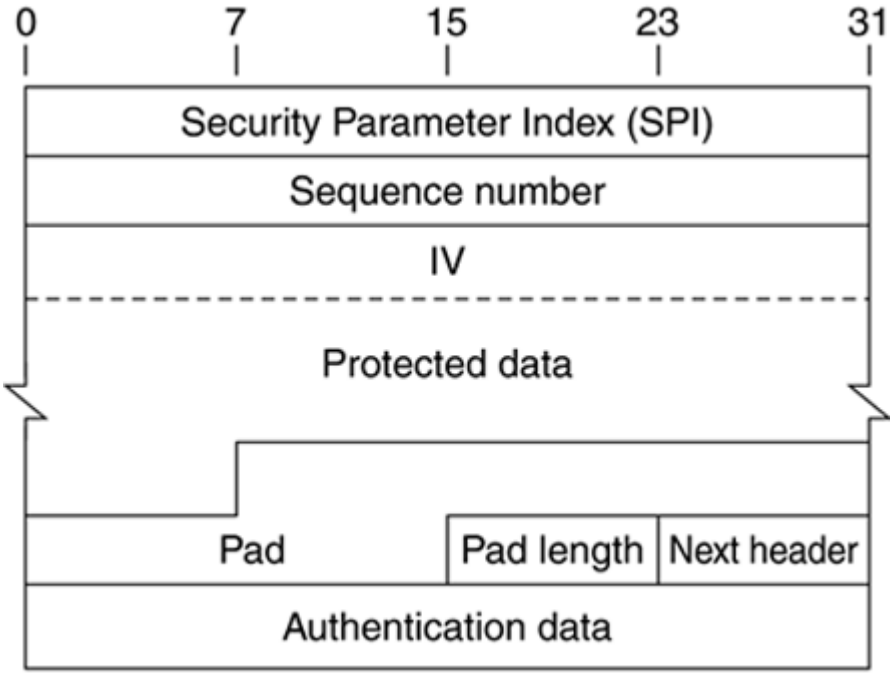


Figure III.9: la forme d'un ESP

- **SPI :** index unique définissant la SA pour ce paquet ;
- **Numéros de séquence :** compteur utile au mécanisme d'anti-répétition ;
- **Données :** donnée du protocole de couche supérieure ;
- **Bourrage :** sert à l'encryptions des données. Certains protocoles nécessitent une certaine taille afin d'être plus efficace et/ou applicable ;
- **Taille du bourrage :** indique la taille du bourrage ;
- **Entête suivante :** ce champ permet de spécifier le type du protocole transporté
- **Données Authentification (variable) :** champs contenant les signatures de hachages permettant d'authentifier l'émetteur et l'authenticité des données. La taille de ce champ dépend des protocoles de hachage et d'encryptions utilisés
- **Le champ bourrage** peut être nécessaire pour les algorithmes de chiffrement par blocs ou pour aligner le texte chiffré sur une limite de 4 octets [34].

Voyons maintenant comment est appliquée la confidentialité dans ESP [35] :

- L'expéditeur encapsule, dans le champ "charge utile" d'ESP, les données transportées par le datagramme original et éventuellement l'en-tête IP (mode tunnel).
- Ajoute si nécessaire un bourrage.
- Chiffre le résultat (données, bourrage, champs longueur et en-tête suivant)
- Ajoute éventuellement des données de synchronisation cryptographiques (vecteur d'initialisation) au début du champ "charge utile" Si elle a été sélectionnée, l'authentification est toujours appliquée après que les données ne soient chiffrées. Cela permet, à la réception, de vérifier la validité du datagramme avant de se lancer dans la coûteuse tâche de déchiffrement. Contrairement à AH, l'authentification dans ESP est appliquée uniquement sur le "paquet" (en-tête + charge utile + trailer) ESP et **n'inclut ni l'en-tête IP ni le champ d'authentification**.
- Le SPI (Security Parameters Index) permet de caractériser l'association de sécurité utilisée pour la communication (SA).
- Les données d'authentification contiennent la valeur de vérification d'intégrité (ICV)

permettant de vérifier l'authenticité des données du paquet.
- Le numéro de séquence pour éviter le rejeu.
- Les données chiffrées sont contenues dans la partie « champ libre » (ou Payload Data) du paquet.
- Le champ En-tête suivant (Next Header) indique la nature des informations contenues dans le Payload Data (champ libre).

Transport Mode (ESP) :

Lorsque nous utilisons le mode de transport, nous utilisons l'en-tête IP d'origine et insérons un en-tête ESP. Voici à quoi cela ressemble (**Figure III.10**) :

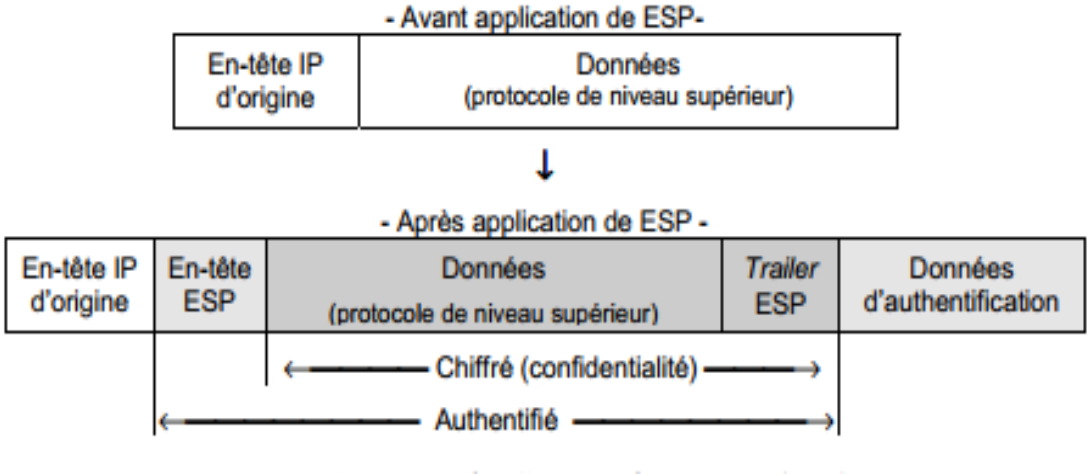


Figure III.10: position de ESP en mode transport (ipv4)

Tunnel Mode (ESP) :

C'est là que nous utilisons un nouvel en-tête IP qui est utile pour les VPN de site à site la (**Figure III.11**):

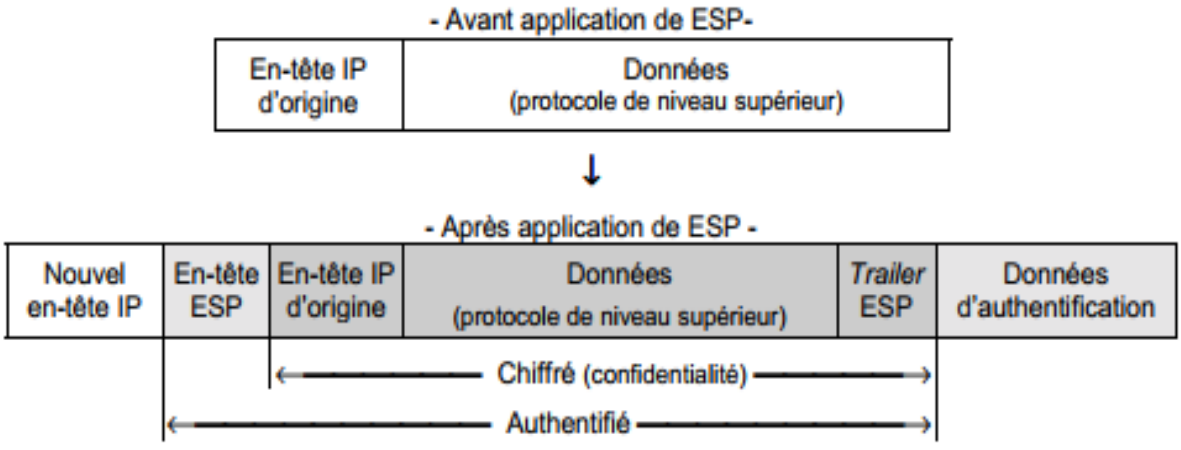


Figure III.11: position de ESP en mode tunnel (ipv4)

L'un ou l'autre des protocoles (AH et ESP) peut être utilisé seul pour protéger un paquet IP, ou les deux protocoles peuvent être appliqués ensemble (imbriqué) au même paquet IP.

III.8 Les algorithmes et protocoles IPSec :

III.8.1 Algorithmes de chiffrement :

Les algorithmes de chiffrement protègent les données afin qu'elles ne puissent être lues par un tiers lors de leur transfert. IPsec prend en charge au minimum trois algorithmes de chiffrement :

- **DES (norme de chiffrement de données) :**
  Utilise une clé de chiffrement de 56 bits. Il s'agit du plus faible des trois algorithmes.
- **3DES (Triple DES) :**
  Un algorithme de chiffrement basé sur la norme DES. Cet algorithme utilise trois fois la norme DES pour chiffrer des blocks de données de taille multiple à 64bits avec une clé de 112 ou 168 bits.
- **AES (norme de chiffrement avancée) :**
  L'algorithme de chiffrement le plus puissant disponible. IPsec peut utiliser les clés de chiffrement AES suivantes : 128, 192 ou 256 bits.

III.8.2 Algorithmes d'authentification :

Les algorithmes d'authentification vérifient l'intégrité des données et l'authenticité d'un message. IPsec prend en charge trois algorithmes d'authentification :

- **HMAC-MD5 (**Algorithm 5) **:** produit un prétraitement de message de 128 bits (16 octets), ce qui le rend plus rapide que SHA-1 et SHA2. C'est l'algorithme le moins sécurisé.
- **HMAC-SHA1** (Algorithm 1) : produit un prétraitement de message de 160 bits (20 octets). Bien qu'il soit plus lent que le MD5, ce prétraitement plus grand offre plus de résistance aux attaques en force.
- **HMAC-SHA2** (Algorithm 2) **:** SHA2 est plus sécurisé que SHA1 ou MD5**.**

III.9 Gestion des clefs IPSec :

III.9.1 Les différents types de clefs :

- **Clefs de chiffrement de clefs :** Ces clefs sont utilisées afin de chiffrer d'autres clefs et ont généralement une durée de vie longue. Les clefs étant des valeurs aléatoires, l'utilisation d'autres clefs pour les chiffrer rend les attaques par cryptanalyse (tentatives de déchiffrement du message) plus difficiles à leur niveau. La cryptographie à clef publique est souvent utilisée pour le transport de clefs, en chiffrant la clef à transporter à l'aide d'une clef publique [34].
- **Clefs maîtresses :** Les clefs maîtresses sont des clefs qui ne servent pas à chiffrer mais uniquement à générer d'autres clefs par dérivation. Une clef maîtresse peut ainsi être utilisée, par exemple, pour générer deux clefs ; une pour le chiffrement et une pour la signature [34].
- **Clefs de session ou de chiffrement de données :** Ces clefs, contrairement aux précédentes, servent à chiffrer des données.

III.9.2 PKI - Public Key Infrastructure :

De nombreuses applications et protocoles utilisent le cryptage à clef publiques sur d'importants réseaux. Il est nécessaire de pouvoir gérer dans ce cas un nombre important de clefs publiques. Pour cela, on a recours à des Infrastructures à Clefs Publiques, ou PKI (Public Key Infrastructure.

Ces infrastructures se basent généralement sur des autorités de certification (CA : Certificate Authorities), qui garantissent l'authenticité des clefs publiques et permettent une gestion hiérarchisée de celles-ci [34].

III.9.3 Echange de clefs et authentification :

La première étape lors de l'établissement d'une communication sécurisée, est l'authentification des interlocuteurs. Ensuite, un échange de clef permet l'utilisation d'un mécanisme de sécurisation des échanges : l'authentification est ainsi étendue à la suite de la communication. Les types d'échange de clef sont :

A. Les mécanismes de sécurisation des échanges :

- Le « Perfect Forward Secrecy » (PFS) est assurée par une renégociation régulière des clefs. Dans le cas où un attaquant intercepterait et déchiffrerait une clef de session, celle-ci serait probablement déjà « périmée » avant qu'il puisse l'utiliser.
- L'Identity Protection, ou protection de l'identité, est respectée si un message intercepté ne permet pas de déterminer l'identité des tiers communiquant.
- Le Back Traffic Protection consiste en une génération de nouvelles clefs de sessions sans utilisation de clefs maîtresses. Les nouvelles clefs étant indépendantes des clefs précédentes, la découverte d'une clef de session ne permet ni de retrouver les clefs de session passées ni d'en déduire les clefs à venir.

B. Algorithme de Diffie-Hellman :

Inventé en 1976 par Diffie et Hellman, ce protocole permet à deux tiers de générer un secret partagé sans avoir aucune information préalable l'un sur l'autre. Il est basé sur un mécanisme de cryptage à clef publique, et fait donc intervenir les valeurs publiques et privées des tiers. Le secret généré à l'aide de ce protocole peut ensuite être utilisé pour dériver une ou plusieurs clefs (clef secrète, clef de chiffrement de clefs...).

Cet algorithme est très simple pour l'échange des clefs :

Soient 2 personnes A et B désirant communiquer sans utiliser une clef secrète. Pour cela ils se mettent d'accord sur 2 nombres g et n tels que n soit supérieur à g et g supérieur à 1, et cela sur un canal non sécurisé (il faut que n soit grand : de l'ordre de 512 ou 1024 bits pour que l'échange des clefs soit sécurisé). Ils prennent chacun chez eux un nombre aléatoire.

- A choisi x, calcul $X=g^x \bmod n$ et l'envoie à B ;
- B choisit y, calcul $Y=g^y \bmod n$ et l'envoie à A.

Ainsi le pirate peut intercepter X, et Y mais il lui est très difficile d'en déduire x et y (c'est sur ce principe que repose la sécurité de l'algorithme). Une fois dans son coin,

- A calcule : k=Yx mod n et
- B calcule k'=Xy mod n.

En regardant de plus près, on constate que : **k = k' = gxy mod n**. Ainsi, A et B ont réussi à créer une clef privée dont ils sont les seuls détenteurs.

III.10 Internet Key Exchange :

IKE est un protocole IPsec permettant d'échanger dynamiquement des paramètres et des clés. IKE rend IPsec plus avancé en automatisant la procédure d'échange/de mise à jour des clés nécessaire pour se défendre contre les attaques par mot de passe contre les sessions IPsec. IKE aide à l'établissement automatique d'associations de sécurité (SA) entre deux points de terminaison IPsec. Une SA est un accord de paramètres IPsec entre deux paires [36]. IKE utilise en fait d'autres protocoles pour effectuer l'authentification par les pairs et la génération de clés :

a. ISAKMP :

Le protocole Internet Security Association and Key Management Protocol définit les procédures d'établissement, de négociation, de modification et de suppression des associations de sécurité. Toutes les négociations de paramètres sont gérées via ISAKMP, telles que l'authentification d'en-tête et l'encapsulation de la charge utile (les en-têtes et les modes discutés précédemment). ISAKMP effectue l'authentification par les pairs, mais cela n'implique pas d'échange de clé.

b. Oakley :

Le protocole Oakley utilise l'algorithme Diffie-Hellman pour gérer les échanges de clés entre les SA IPsec. Diffie-Hellman est un protocole cryptographique qui permet à deux points d'extrémité d'échanger un secret partagé sur un canal non sécurisé.

III.10.1 Les phases du protocole IKE :

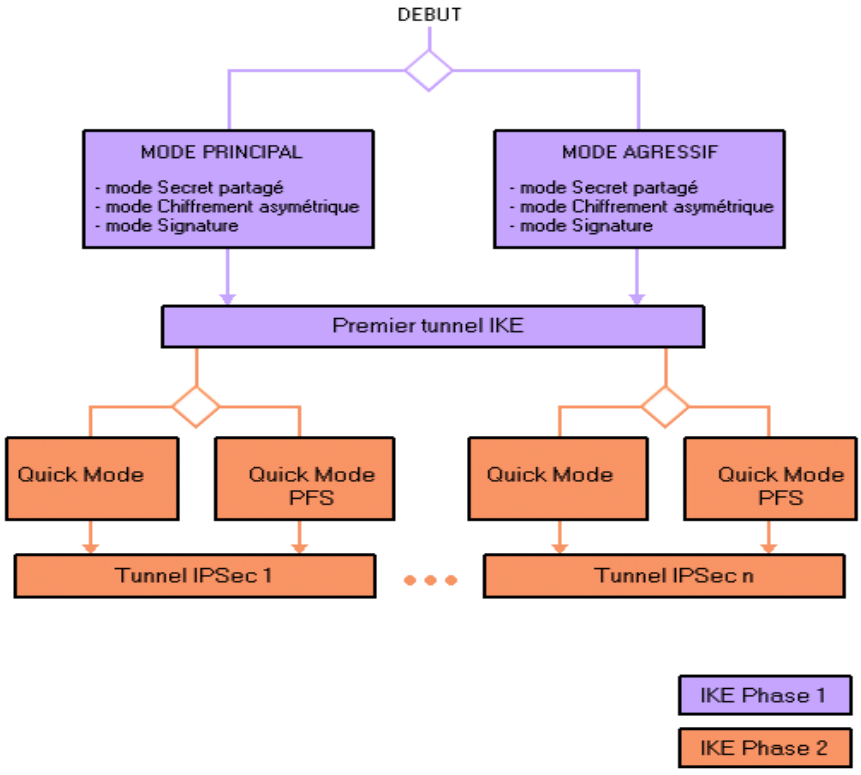


Figure III.12: les phases de protocole IKE

Le protocole IKE comporte deux phases :

La Phase 1 :

Deux modes sont possibles pour la première phase, le mode "main" et mode "agressive". Nous détaillerons le mode main.

Cette phase va servir à la création d'une première clé maitresse qui va permettre par la suite la génération de 3 autres clés dérivant de celle-ci. Cette clé peut être générée selon 3 modes offerts par IKE.

- Le mode **« secret partagé » :** implique que les hôtes partagent déjà un secret qui permettra la mise au point de cette clé.
- Le mode « **Chiffrement asymétrique » :** se base sur les cryptos système à clé publique pour échanger les données sensibles et donc établir le secret partagé.
- Le mode « **Signature** » : quant à lui, se sert du chiffrement asymétrique pour signer et authentifier les hôtes alors que le secret partagé est établi grâce à Diffie-Hellman.

Une fois la première clé générée, elle est dérivée en 3 autres clés qui serviront à la création du tunnel IKE sécurisé entre les hôtes (en faîte SA-ISAKMP). L'une des clés sera utilisée pour l'authentification, l'autre pour le chiffrement et la dernière sera utilisé lors de la phase 2 du protocole. Ce canal, sécurisé, est ensuite utilisé pour la deuxième phase IKE. Plus précisément, lors de cette phase, les échanges permettent de définir l'association de sécurité puis d'établir le secret partagé et enfin d'authentifier les hôtes.

Il faut noter que le mode « agressive » permet de limiter les communications en utilisant certains paramètres d'office. D'autre part, les SA-ISAKMP utilisent un chiffrement (DES ou 3DES) lors de l'échange des clefs de session [37].

La Phase 2 :

L'objectif de la deuxième phase est de créer les tunnels IPSec (SA) pour les échanges effectifs entre les hôtes. Deux SA par hôtes, un pour chaque sens de communication, conservées dans la SAD (Security Association Database). C'est lors de cette phase que chaque hôte donne ses préférences en matière d'algorithme et établissent le matériel cryptographique. Les clés de session sont générées à partir de l'une des clés dérivées, générée durant la phase 1 d'IKE.

Cependant, lorsque le mode «Perfect Secrecy» est utilisé, les hôtes doivent échanger de nouveaux secrets, ceci afin de couper la relation systématique entre les nouvelles clés générés et la clé de la phase 1. Cet échange s'effectue via le protocole d'échange Diffie-Hellman. Cette phase sert aussi à spécifier les échanges devant bénéficier des services IPSec (utilisation de la Security Policy Database) [38].

III.11 Conclusion :

Dans ce chapitre nous avons étudié le fonctionnement et les services de sécurité fournis par le protocole IPsec. IPsec est comme nous l'avons vu est un assemblage de plusieurs protocoles et mécanismes qui permet d'assurer la protection des données échangées sur le réseau.

Dans le chapitre qui suit nous allons sécuriser de protocole de routage AODV la protection des paquets de contrôle à l'aide de protocole AH (Authentication header) en mode transport, ce qui fournit l'intégrité et l'authentification et on va comparer le fonctionnement du protocole AODV avec notre approche dont le routage est protégé par le protocole AH, à l'aide de le simulateur Omnet++.

.

Chapitre IV : Etude de la simulation

IV.1 Introduction :

Ce chapitre a pour objectif de comparer le fonctionnement du protocole de routage AODV, en termes de temps de découverte de routes en deux cas ;
En utilisant du le protocole AODV ordinaire sans le sécuriser, ensuite utiliser AODV avec le protocole IPSec (protocole AH) afin de sécuriser les paquets de routage et enfin évaluer les performances de protocole sélectionné pour l'étude. Dans ce chapitre, nous allons présenter principalement les outils utilisés pour réaliser notre simulation, puis les différentes étapes permettant de mettre en œuvre le protocole à l'aide de simulateur OMNET et du INET Framework, et enfin, des présentations détaillées des résultats obtenus et la discussion de ces résultats.

IV.2 Choix du simulateur :

Le déploiement d'un réseau Adhoc exige une étape de simulation avant son installation sur site. La simulation permet de tester à moindre coût les performances d'une solution.

OMNeT++ est un environnement de simulation à évènements discrets basé sur le langage C++, une application open source. Il est totalement programmable, paramétrable et modulaire grâce à son architecture flexible et générique, il a été utilisé avec succès dans divers domaines.

OMNeT++ sera notre environnement de simulation, grâce à son architecture et sa conception de modèles se rapprochant de la réalité.

IV.3 Présentation OMNET ++ :

IV.3.1 Définition :

Développé par András Varga , Objective Modular Network Test-bed in C++ (OMNeT++) est un simulateur libre à événements discrets. Il est basé "composants" et propose un Framework de simulation modulaire. OMNeT++ était initialement conçu pour simuler les communications réseaux. Cependant, grâce à son architecture générique et flexible, il est actuellement utilisé avec succès dans plusieurs autres champs tels que la simulation de systèmes complexes de traitement et de communications de données, la simulation de réseaux de files d'attente, la modélisation des systèmes multiprocesseurs et pas mal d'autres systèmes distribués.

OMNET ++ est un environnement de simulation orienté objet pour événements discrets basé sur C++ qui peut être utilisé pour simuler des réseaux, des systèmes multiprocesseurs et d'autres systèmes discrets. OMNET ++ est largement utilisé dans une variété d'applications grâce à son architecture modulaire, y compris la modélisation de protocoles de communication, la modélisation de réseaux et la modélisation de réseaux sans fils. En général, il peut être utilisé pour tout système à événements discrets qui peut être conçu autour d'entités de communication envoyant des messages.

OMNET++ fournit des outils pour la création et la configuration des modèles de réseaux (les fichiers NED et INI) et des outils pour l'exécution des programmes ainsi que pour l'analyse des résultats de simulation [39].

Figure IV.1: Le lancement du simulateur OMNET++

IV.4 Architecture de OMNET++ :

Les modèles OMNET++ constituent en un ensemble de modules hiérarchiquement emboités tel qu'il est montré dans la **(Figure IV.2) :**
L'architecture d'OMNET++ est hiérarchique composé de modules. Un module peut être soit module simple ou bien un module composé. Les feuilles de cette architecture sont les modules simples qui représentent les classes C++. Pour chaque module simple correspond un fichier .cc et un fichier.h. Un module composé est composé de simples modules ou d'autres modules composés connectés entre eux. Les paramètres, les sous modules et les ports de chaque module sont spécifiés dans un fichier.ned. La communication entre les différents modules se fait à travers les échanges de messages.
Les messages peuvent représenter des paquets, des trames d'un réseau informatique, des clients dans une file d'attente ou bien d'autres types d'entités en attente d'un service. Les messages sont envoyés et reçus à travers des ports qui représentent les interfaces d'entrer et de sortie pour chaque module. La conception d'un réseau se fait dans un fichier .ned et les différents paramètres de chaque module sont spécifiés dans un fichier de configuration (.ini). OMNET++ génère à la fin de chaque simulation deux nouveaux fichiers omnet.vec et omnet.sca qui permettent de tracer les courbes et calculer des statistiques [40] .

- ✓ La modélisation des protocoles de communications
- ✓ La modélisation des réseaux filaires et sans fils
- ✓ La modélisation des systèmes répartis
- ✓ L'architecture Hardware

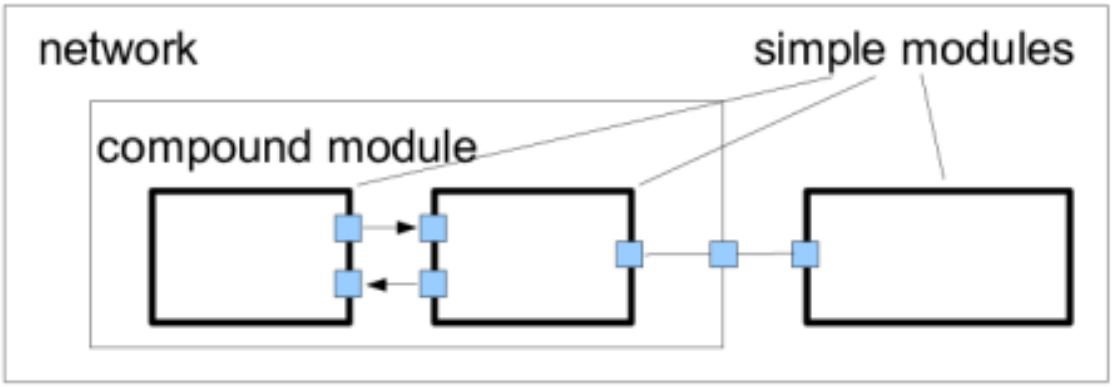


Figure IV.2: Architecture modulaire du simulateur OMNET++

L'utilisateur définit la structure du modèle (les modules et leurs interconnexions) par le biais du langage NED. Les composants d'une description NED sont les déclarations des modules simples, les définitions des modules composés et des réseaux.

La déclaration d'un module simple décrit les interfaces de ce module et ses paramètres Les Définitions des modules composés comportent la déclaration des interfaces externes du module (les portes et les paramètres), et la définition de ses sous-modules et leurs interconnexions. La définition d'un réseau est un module composé qualifié comme un modèle de simulation autonome [41].

IV.5 Installation d'OMNeT++ :

- Télécharger le code source OMNeT ++ à partir de **http://omnetpp.org**. Assurer de sélectionner l'archive spécifique à Windows, nommée **omnetpp-5.2-src-windows.zip**.
- Le paquet est presque autonome : en plus des fichiers OMNeT ++, il comprend un compilateur C ++, un environnement de génération de ligne de commande et toutes les bibliothèques et programmes requis par OMNeT ++.
- Copier l'archive OMNeT ++ dans le répertoire où nous souhaitons l'installer, choisir un répertoire dont le chemin complet ne contient aucun espace ; Par exemple, ne pas mettre OMNeT ++ sous Program Files.
- Extraire le fichier téléchargé, un répertoire nommé omnetpp-5.2 est créé [42].

IV.5.1 Les principaux fichiers d'OMNET++ :

OMNET++ est composé par différents principaux fichiers [43] sont :

Fichier (.NED):

Utilise le langage NED (Network Descriptor) de description de réseaux. Il peut être utilisé en deux modes : Mode Graphique ou Mode Texte qui permettent de décrire les paramètres et les ports du module. Les erreurs commises sont indiquées en temps réel par un point rouge situé à la gauche du code Un exemple de fichier Ned en mode "source" & "Graphique" sont présentés dans la **(Figure IV.3)** et **(Figure IV.4).**

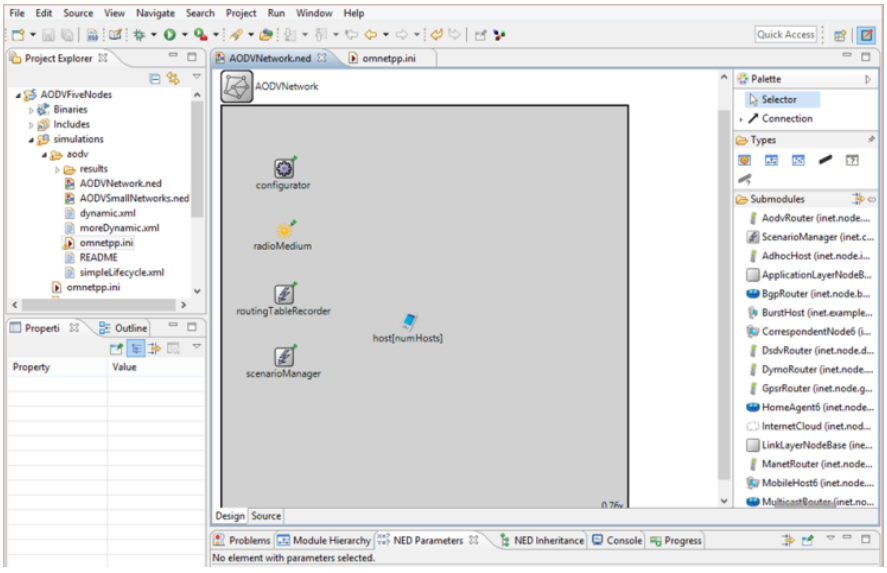


Figure IV.3: Ficher Ned en mode graphique

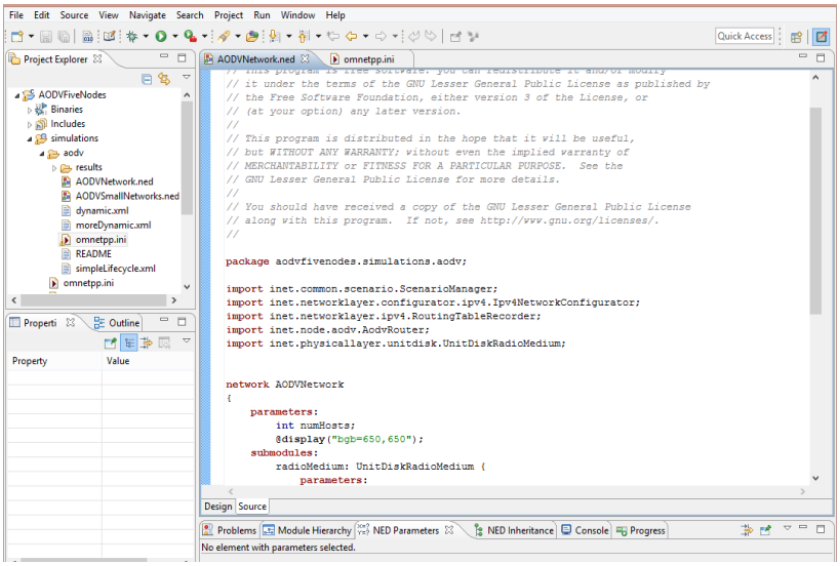


Figure IV.4: Fichier Ned en mode texte

Fichier(.ini) :

Est lié étroitement avec le fichier NED. Permet à l'utilisateur d'initialisé les paramètres des différents modules ainsi la topologie du réseau. Voici un exemple présenté ci-dessous :

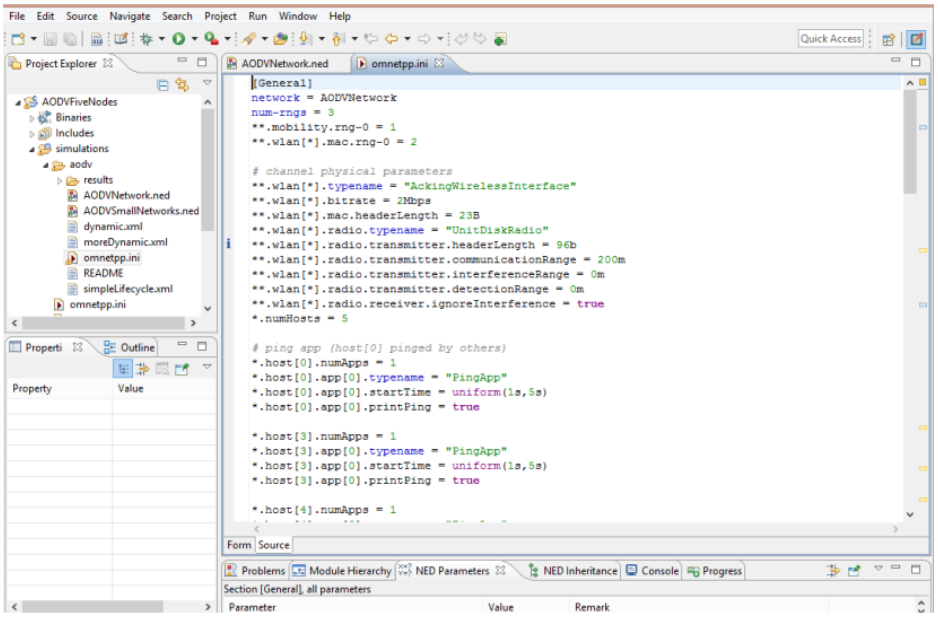


Figure IV.5: Exemple d'un fichier *.ini

Fichier(.msg) :

Les modules communiquent en échangeant des messages. C'est dernier peuvent être déclarés dans un fichier dont l'extension est(.msg) ou l'on peut ajouter des champs de données. OMNET++ traduira les définitions de messages en classes C++ le diagramme suivant peut donner une idée plus détaillée sur le développement d'exécution d'une simulation sous OMNET++

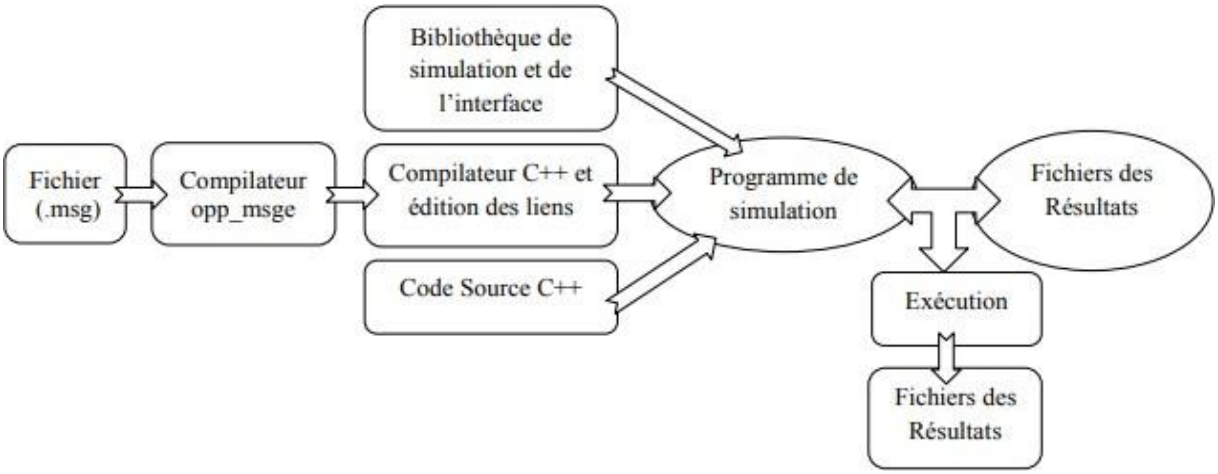


Figure IV.6: Exécution d'une simulation sous OMNeT++

Structure d'un nœud mobile dans OMNET++ :

Dans OMNET++, un nœud mobile a une structure représentée par (**Figure IV.7)** et **(Tableau 1)** :

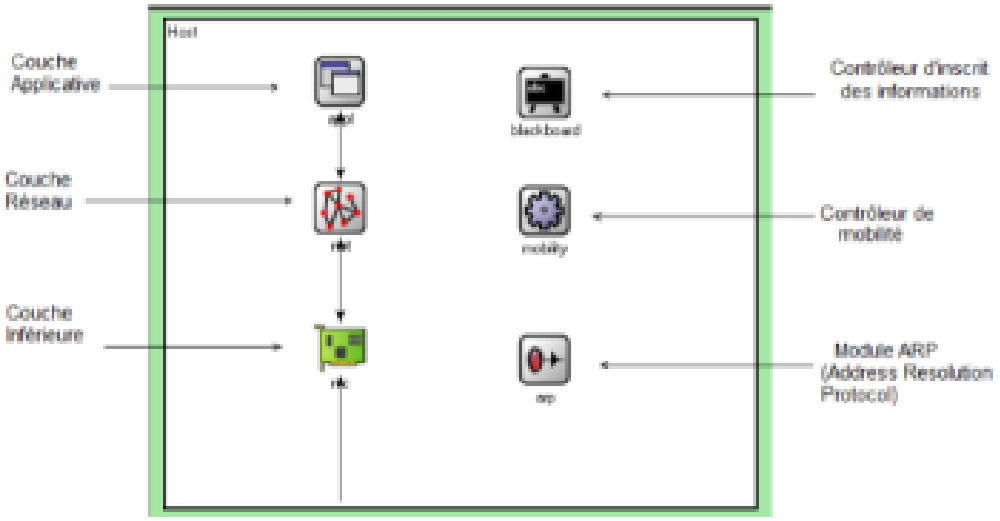


Figure IV.7: Structure d'un nœud mobile dans OMNET++

Application	FTP, Telnet, générateur de trafic (IPTrfGen..), Ethernet, Ping App, UDPApp, TCPApp
Transport	TCP, UDP, RTP
Réseau	IPv4, IPv6, ARP, OSPF, LDP, MPLS, ICMP, TED...
Liason	Mgmt, MAC, Radio
Node	Ad Hoc, Wireless, MPLS...

Tableau IV.1: Structure d'un nœud mobile dans OMNET++

IV.6 INET Framework :

INET est une librairie open source pour la simulation des réseaux informatiques dans l'environnement OMNeT++. Elle contient IPv4, IPv6, TCP, UDP, des protocoles implémentés, et plusieurs modèles d'application [44].

Elle contient actuellement des modèles de protocoles pour la suite TCP/IP (IPv4, IPv6, TCP, SCTP, UDP, …), des modèles de la couche liaison pour les réseaux filaires et sans fils (Ethernet, PPP, IEEE802.11, …), des modèles MPLS avec signalisation RSVP et LDP, un support à la mobilité et plusieurs autres protocoles et composants.

Ses modules sont organisés dans des paquetages qui sont à leurs tours organisés selon les couches du modèle OSI (exemple : inet.applications, inet.transport, …).

Du point de vue architectural, INET respecte le concept modulaire d'OMNeT++ : les protocoles sont représentés par des modules simples dont les interfaces externes sont décrites par des fichiers NED et le comportement est implémenté à l'aide de classes C++. Les nœuds sont construits par composition de plusieurs modules simples.

D'autres modules (qui n'implémentent pas de protocoles) sont utilisés pour assurer des tâches spécifiques au cours de la simulation : On en trouve côté nœud, le module InterfaceTable qui contient la table des interfaces réseau (eth0, wlan0, …), les tables de routage RoutingTable4 et RoutingTable6 pour IPv4 et IPv6 (respectivement) et le module NotificationBoard qui facilite la communication entre les différents modules.

Au niveau réseau, on cite le module FlatNetworkConfigurator qui sert à attribuer les adresses IP aux différents nœuds et de configurer un routage statique, le module ScenarioManager qui contrôle les expériences de simulation et la planification d'évènements et le module ChannelControl requis pour les simulations sans fil et permet de garder la trace des nœuds à l'intérieur d'une zone d'interférences avec d'autres nœuds.

En ce qui concerne l'interaction entre ses différents éléments, INET gère la communication entre les différentes couches de protocoles via un processus d'encapsulation/décapsulation avec ControlInfo comme un objet attaché au message pour véhiculer une information additionnelle à la couche prochaine.

Un mode d'appel direct est suivi pour lier les autres modules, souvent en communication. Cela, est assuré par son module NotificationBoard qui joue le rôle d'intermédiaire entre le module où les évènements apparaissent et les modules qui sont intéressés par ces évènements. Son fonctionnement est basé sur le concept publication/abonnement selon lequel les modules peuvent s'abonner à des catégories de changements (exemple : un tableau de routage change d'état, un canal de communication devient libre). Quand l'un des changements se produit, le module hôte (exemple : Table de routage, couche physique) informe le module NotificationBoard, qui à son tour, diffuse l'information vers tous les modules à cette catégorie de changement.

INET constitue aujourd'hui un framework incontournable pour OMNeT++. La richesse de ses modèles et la réutilisabilité de ses composants lui ont garanti un large déploiement chez la communauté OMNeT++ et lui ont permis d'être le socle sur lequel se basent plusieurs extensions telle que CoRE4INET l'extension INET implémentant le protocole AODV [45].

Catalogue de modèles :

Les composants de modèle suivants [46] (protocoles, applications et autres modèles) sont disponibles pour INET Framework :

	Protocol	Projet
Application	CBR/VBR , HTTP, File Transfer , DHCP…	INET
Transport	TCP,UDP,SCTP,RTp,RTCP	INET
Reseaux	IPv4, ICMPV4 ,ARP ,IGMPv3 ,IPv6	INET
Routage	link-state routing , OSPFv2(1) , OSPF(2) , BGPv4 , BGP(2) ,RIP	INET
Manet Routage	AODV , DYMO , GPSR , DSDV, DSR,OLSR	INET
Fils	PPP, Ethernet, STP ,RSTP ,TTE ,802.1avb ,EPON , TDM/WDm-PON	INET
Sans fils	802.11 , 802.11p , 802.1e ,802.15.4, LTE(User-Plane) ,LTE(Control-Plane)	INET
Mobility/Environment	Various Mobility Models	INET

Tableau IV.2: La liste des principaux composants de modèle disponible dans INET FW

IV.7 Les Avantages et Les Inconvénients

Avantages :

- Architecture modulaire permettant l'intégration de nouveau modèle.
- Utilisation du C++ pour le développement du noyau.
- Les classes de base du simulateur peuvent être étendue et personnalisées.
- Conception de modèle rapprochant de la réalité.
- La mise en route avec ce simulateur est assez simple grâce à une conception claire du simulateur [47].
- Il fournit également une puissante bibliothèque d'interfaces graphiques pour l'animation et la gestion du débogage.

Inconvénients :

- Peu de modèles pour les réseaux sans fil
- Il y a un manque cruel de protocoles disponibles dans la bibliothèque comparée à d'autres simulateurs [47]

IV.8 Partie De La Simulation

Nous avons évoqué les différentes attaques auxquelles le protocole de routage AODV peut être soumis ; Pour protéger ce protocole, le chemin entre les nœuds (émetteur / intermédiaire / récepteur) doit être protégé. Puisque l'émetteur n'a pas de route vers la destination il suffit de sécuriser les liaisons entre les voisins ainsi les routes possibles se trouvent protégées par notre approche.

Dans un souci de sécurisation de ces chemins, nous avons suggéré d'utiliser IPSec qui permet de protéger les échanges IP et AODV et précisément on va utiliser son protocole de base AH qui fournit l'intégrité et l'authentification. AH authentifie les paquets de routage en les signant, ce qui

assure l'intégrité de l'information. Une signature unique est créée pour chaque paquet envoyé ainsi empêche que l'information soit modifiée.

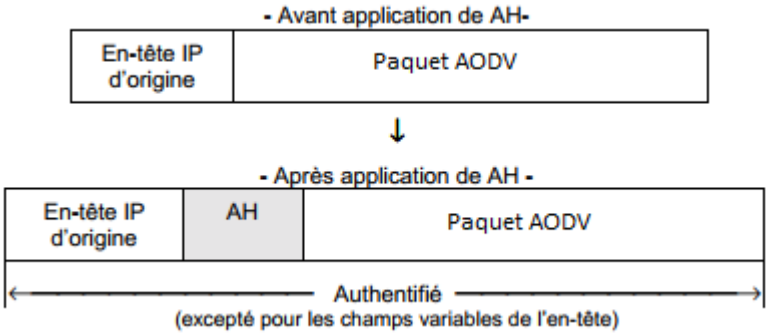


Figure IV.8 : Encapsulation du paquets AODV dans AH

Pour étudier l'effet de l'utilisation le protocole de base d'IPsec (AH) sur le fonctionnement du protocole de routage AODV, En termes de **temps de découverte de routes** (Le Temps entre la première RREQ envoyé par un nœud source et la RREP correspondante reçu depuis le nœud destination), on prend comme paramètres de scenario deux éléments : la mobilité et la densité ;

La mobilité est la vitesse des nœuds des réseaux, et la densité est le nombre des nœuds dans le scénario. On va créer quatre scenarios sur Omnet++, dans les deux premier on va simuler le réseau sans l'utilisation d'AH (l'un avec 5 nœuds et l'autre avec 10 nœuds) avec une gamme de vitesses variées qui reflète les applications des réseaux Adhoc ; Dans les deux derniers on va appliquer le protocole AH dans les paquets de contrôle sur les mêmes scenarios précédents, ensuite comparer les résultats.

Les paramètres de simulation utilisés dans ces deux scénarios sont :

Nombre de nœuds	5, 10
Le temps de simulation	500s
BitRate	2Mbps
Transmission range	200m
surface de simulation	1000m * 1000m
Le Protocole de routage	AODV
Interfaces de routage	Wlan
Traffic type	UDPAppBasic
Placement des nœuds	Uniforme

Tableau IV.3: Les paramètre de la simulation

Section static du fichier omnet.ini représenté par :

```
# lifecycle
**.hasStatus = true

[Config Static]
**.host[*].mobility.typename = "MassMobility"

**.aodv.activeRouteTimeout = 3s
**.aodv.ttlStart = 10
**.host[*].mobility.changeInterval = normal(5s, 0.1s)
**.host[*].mobility.angleDelta = normal(0deg, 30deg)
```

IV.8.1 Topologie en 5 nœuds :

Dans ce réseau, trois nœuds source (S) envoient des données aux nœuds (D),

La destination des nœuds d'envoi :

```
*.host[0].app[0].destAddresses = "host[1](ipv4)"
*.host[3].app[0].destAddresses = "host[2](ipv4)"
*.host[4].app[0].destAddresses = "host[3](ipv4)"
```

Et comme il n'y a pas de chemin établi préalablement entre le nœud source et le nœud destination dans chaque cas, cela nécessite de lancer le processus de découverte de routes par la diffusion des nœuds (S) d'un message de demande de route (RREQ) à ses voisins, y compris le numéro de séquence de cette destination.
La demande de route est inondée en quelque sorte par le réseau jusqu'à ce qu'elle atteigne un nœud qui a une route à la destination ou la destination elle-même, comme illustré dans la figure suivante :

Figure IV.9: Lancement de la simulation

Une fois la simulation terminée, nous avons calculé le temps moyen entre la première RREQ envoyé par nœud source (S) et la RREP reçu depuis le nœud destination (D) pour chaque nœud dans et chaque vitesses (1.25, 23, 45, 100) m/s. Ce qui est illustré dans la figure suivante :

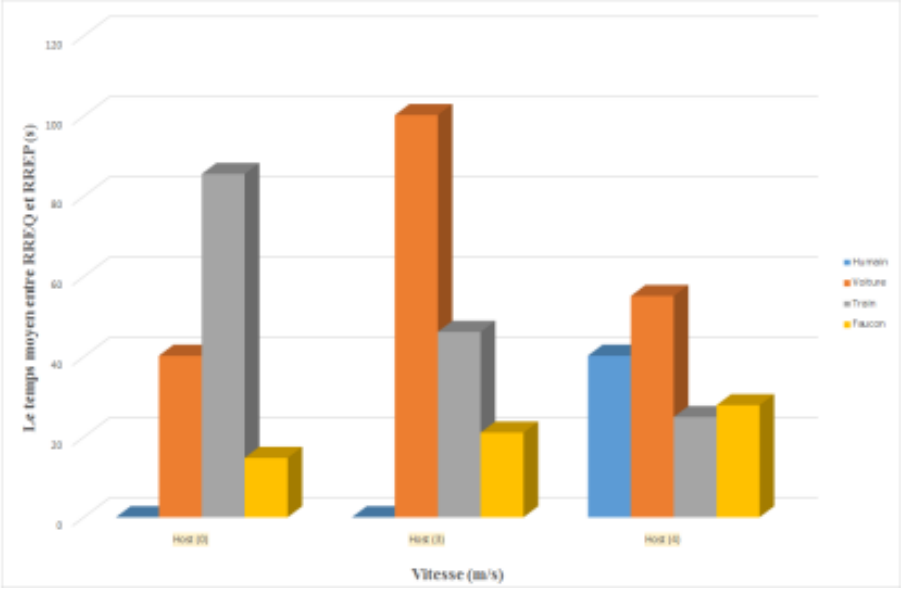


Figure IV.10: Le temps moyen entre RREQ et RREP pour chaque nœud en (s)

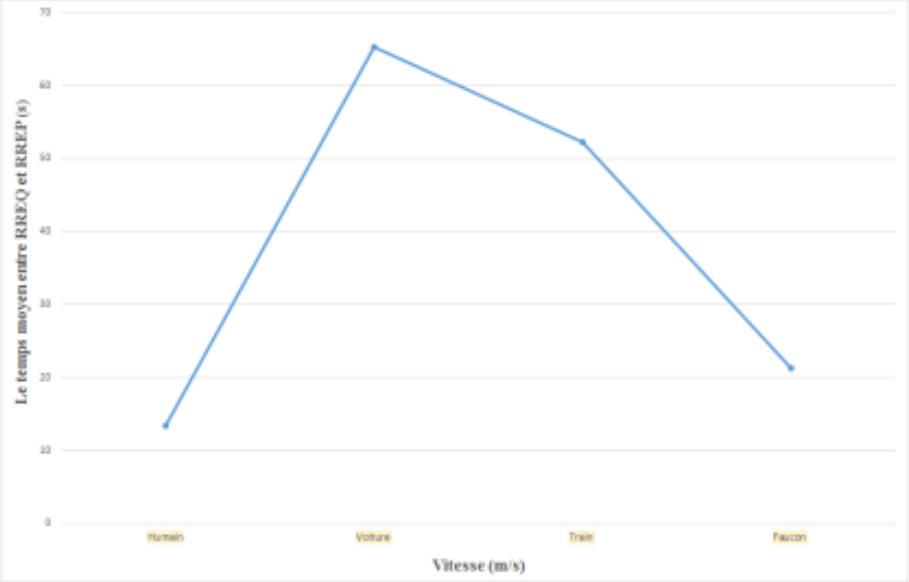


Figure IV.11: Le temps moyen entre RREQ et RREP Pour chaque Vitesse (m/s)

Dans le cas ‘’Humain ‘’, la surface de simulations est grande et la vitesse de déplacement des nœuds est faible, c’est difficile pour Host (0) et Host (3) de trouver un chemin vers leurs destinations durant le temps de la simulation. Dans ce scénario Host (4) est le seul à avoir reçu une route reply (RREP) depuis sa destination Host (3). Ce qui explique le faible temps de découverte de route.

Et dans le cas de ‘’ Voiture’’ la vitesse de déplacement des nœuds est supérieure par rapport à la vitesse ‘’Humain’’. En raison de la vitesse de la voiture, les nœuds peuvent se rapprochés dans plusieurs cas puis ils se détournent, ce qui permet de construire plusieurs routes durant la simulation cela explique l’augmentation, du temps moyen entre RREQ et la RREP comme illustré dans la figure (3).

Au contraire dans le cas de ‘’Train’’ et ‘’ Faucon’’ leurs grande vitesse de mouvements a donnée de la place pour faire converger les nœuds rapidement comme le montre la figure (3), c'est ce qu'il explique la diminution de temps moyen entre RREQ et RREP pour le "Train" et le "Faucon".

Avec l’utilisation d’AH :

Ensuite, dans le même scénario, nous appliquons IPSec, spécifiquement le protocole AH en mode transport. Dans AH, seule la charge utile du paquet IP, précisément le paquet AODV encapsulé dans ce datagramme IP, sont authentifiées et intègre. Le reste du paquet IP reste inchangé, car dans le protocole AODV tous les nœuds sont à la fois routeurs et terminaux. Ceci est appliquer pour chaque voisin séparément.

Si (S) va envoyer paquets de routage à (D) et (I) est un nœud intermédiaire, alors dans ce cas la liaison entre (S) et (I) doit être sécurisé et de même la liaison entre (I) et (D) doit aussi être sécurisé, ainsi le chemin entre (S) et (D) devient systématiquement sécurisé. Dans ce cas, les paquets seront protégés de bout en bout.

Comme nous savons, l’en-tête d’authentification IP (AH) est utilisé pour assurer l’intégrité sans connexion, l’authentification de l’origine des données pour les datagrammes IP et fournir une protection contre la rediffusion.

La première étape pour protéger l'intégrité consiste à créer des hachages à l'aide d'un algorithme de hachage à clé, également connu sous le nom d'algorithme de code d'authentification de message (MAC). L'algorithme de hachage standard génère un hachage basé sur un message, tandis que l'algorithme de hachage à clé génère un hachage basé à la fois sur le message et la clé secrète partagée entre les voisins.

Le hachage est ajouté au paquet RREQ ou RREP et ce dernier est envoyé aux voisins un par un. A la réception du paquet de contrôle, le voisin recrée le hachage avec la clé partagée et s'assurer que les deux hachages correspondent, offrant ainsi une protection de l'intégrité du paquet. La fonction de hachage SHA offre plus de sécurité que MD5. Dans cette simulation, nous avons utilisé l'algorithme de code d'authentification de message de hachage (HMAC-SHA-1), qui effectue deux hachages liés.

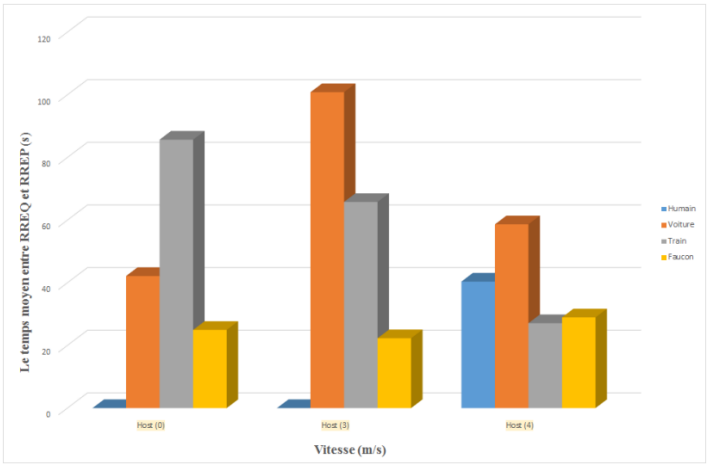


Figure IV.12: Le temps moyen entre RREQ et RREP Pour chaque nœud en (s) (Avec AH)

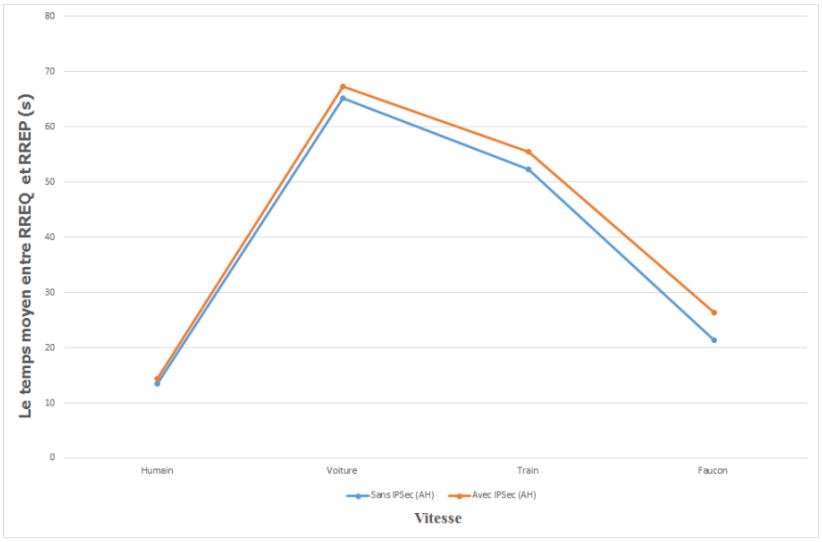


Figure IV.13: Le temps moyen entre RREQ et RREP pour chaque Vitesse (m/s)

On remarque dans ce graphe que le temps moyen entre RREQ et RREP avec l'utilisation d'AH augmente par rapport au cas de routage sans l'utilisation d'AH, par un taux différent d'une vitesse à l'autre.

À mesure que la vitesse de déplacement des nœuds augmente, la différence du temps moyen entre RREQ et RREP dans le cas où le protocole AH s'applique et le cas ordinaire augmente.

IV.8.2 Topologie en 10 nœuds :

Nous appliquerons les mêmes étapes précédentes à un réseau de 10 nœuds Dans ce réseau, Cinque nœuds (S) envoient des données aux nœuds (D),

La destination des nœuds d'envoi : (Avec IPsec et Sans IPSec)

```
*.host[2].app[0].destAddresses = "host[0](ipv4)"
*.host[3].app[0].destAddresses = "host[6](ipv4)"
*.host[4].app[0].destAddresses = "host[7](ipv4)"
*.host[8].app[0].destAddresses = "host[1](ipv4)"
*.host[9].app[0].destAddresses = "host[4](ipv4)"
```

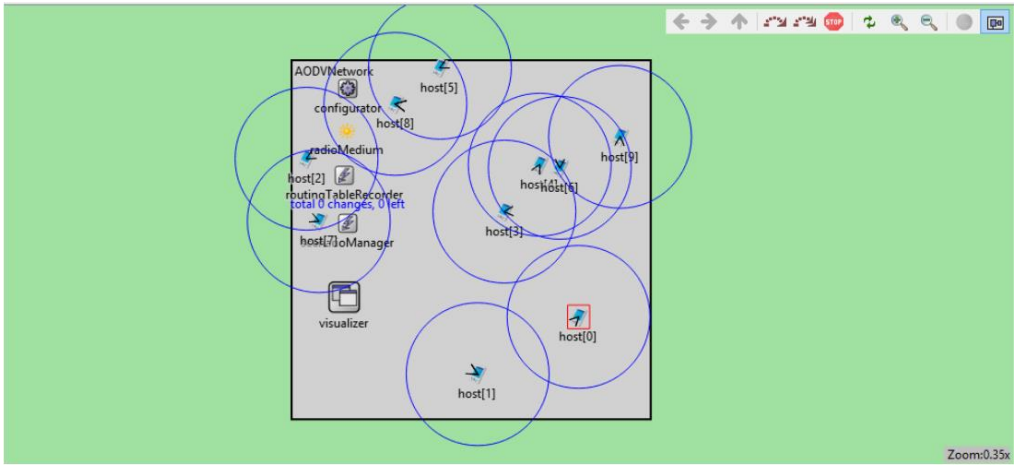


Figure IV.14: Topologie du réseau avant lancement de la simulation

Dans le premier scénario, le réseau n'est pas protégé, ce qui signifie que le protocole AH n'est pas appliqué.

Alors que dans le second scénario, le protocole AH sera appliqué aux messages de contrôle des nœuds du protocole AODV.

Et comme il n'y a pas de chemin entre les nœuds source et de destination dans chaque cas, cela nécessite de découvrir un chemin par la diffusion des nœuds (S) d'un message de demande de route à ses voisins.

Encore lorsque la simulation se termine, nous avons calculé le temps moyen entre la première RREQ envoyé par un nœud (S) et la première RREP qui a reçu depuis le nœud (D) pour chaque nœud dans les quatre vitesses (1.25, 23, 45, 100) m/s qui est illustré dans la figure suivante :

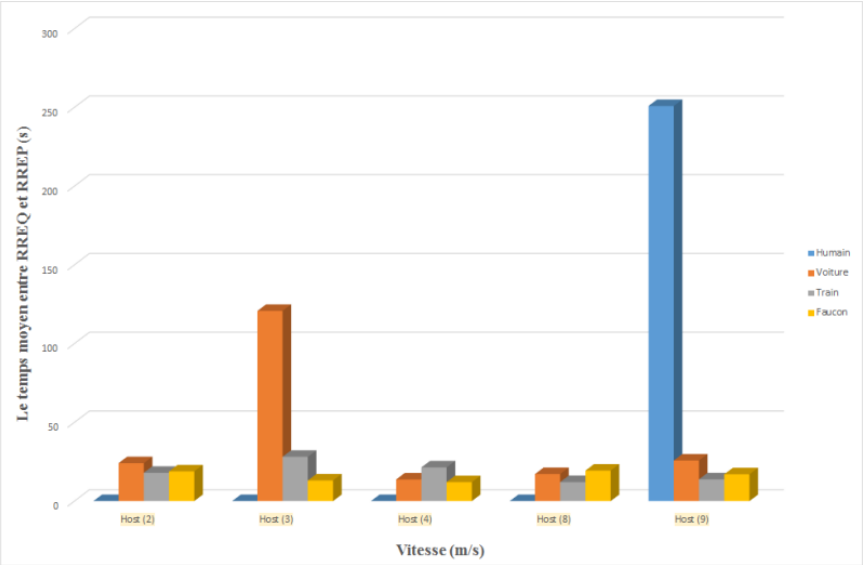


Figure IV.15: Le temps moyen entre RREQ et RREP Pour chaque nœud en (s)

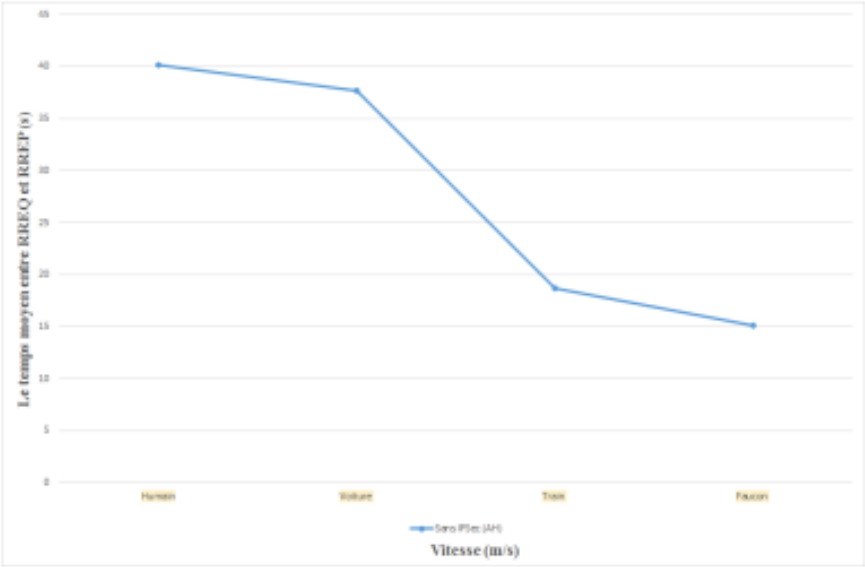


Figure IV.16: Le temps moyen entre RREQ et RREP Pour chaque Vitesse (m/s)

On remarque qu'au fur et à mesure que la vitesse de déplacement des nœuds augmente, le temps moyen entre RREQ et RREP diminue.

Puisque le réseau est dense les routes sont construites fréquemment ce qui augmente le temps de découverte de routes par rapport au scénario de 5 nœuds.

Avec l'utilisation d'AH :

Ensuite, on applique le protocole AH sur ce réseau.

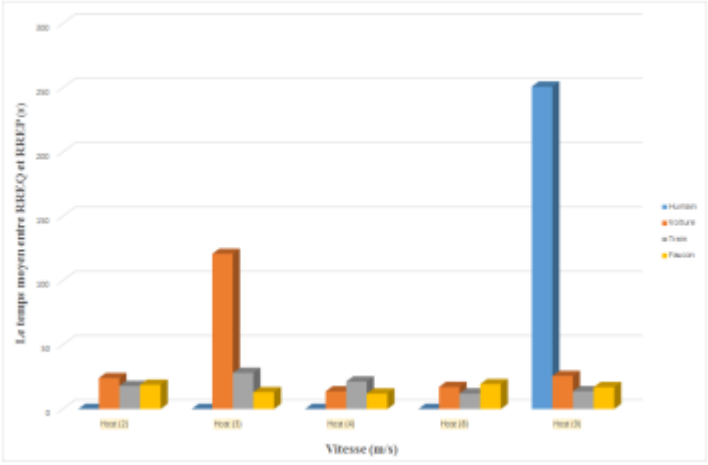


Figure IV.17: Le temps moyen entre RREQ et RREP Pour chaque nœud en (s) (Avec AH)

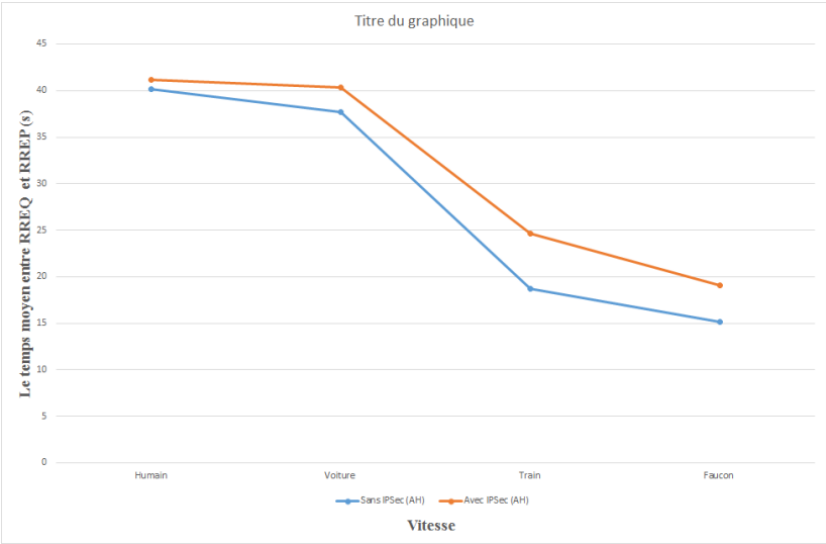


Figure IV.18: Le temps moyen entre RREQ et RREP Pour chaque Vitesse (m/s)
(Avec IPsec et Sans IPSec)

On remarque dans ce graphe que le temps moyen entre RREQ et RREP avec l'utilisation d'AH a augmenté par rapport au cas de routage sans l'utilisation d'AH, par un taux différent d'une vitesse à l'autre.

À mesure que la vitesse de déplacement des nœuds augmente, la différence du temps moyen de découverte de route dans le cas où le protocole AH s'applique et le cas normal (sans AH) augmente.

La différence du temps moyen entre RREQ et RREP d'une vitesse à l'autre dans le cas où le protocole AH s'applique et le cas dont n'est pas appliqué dans le de réseau à 5 nœuds, est inférieur par rapport au réseau de 10 nœuds.

Le tableau suivant représente le nombre total de sauts survenus dans le réseau pendant le routage pour chaque vitesse dans les deux scénarios précédents :

	Host(0)	Host(3)	Host(4)	Total
Humain	0	0	3	3
Voiture	8	4	11	23
Train	22	11	20	53
Faucon	25	21	22	68

Tableau IV.4: Le nombre total de sauts (5 nœuds)

	Host(2)	Host(3)	Host(4)	Host(8)	Host(9)	Total
Humain	0	0	0	0	10	10
Voiture	14	2	21	18	2	52
Train	26	13	23	25	40	127
Faucon	20	27	28	24	21	120

Tableau IV.5: Le nombre total de sauts (10 nœuds)

Si nous comparons les tableaux avec les courbes citées ci-dessus, le temps moyen de découverte de route pour chaque vitesse (m/s) (Avec IPsec et Sans IPSec) pour les réseaux de 5 et 10 nœuds nous remarquons que : plus le nombre total de sauts de chaque vitesse est élevé, plus la différence du temps moyen entre RREQ et RREP augmente d'une vitesse à l'autre.

IV.9 Conclusion

En raison des nombreuses applications sensibles prises en charge par les réseaux Adhoc, la cohérence et l'intégrité du réseau doivent être assurées. Cependant, il est difficile d'assurer une sécurité complète dans un tel réseau si les nœuds sont mobiles et si le réseau est dense. L'implémentation IPSec proposée tente d'assurer la sécurité de la transmission des paquets de contrôle. Par rapport à l'envoi et à la réception de paquets de contrôle sans le protocole AH, l'envoi et la réception de paquets de contrôle avec le protocole AH prennent plus de temps. Cette amélioration varie d'un scénario à l'autre. Sur la base des résultats de la simulation, nous avons conclu que plus le nombre total de nœuds dans le réseau est élevé, plus le temps moyen entre RREQ et RREP est long lorsque le protocole AH est utilisé. Cette augmentation se situe entre 0,5 et 1,5 %, selon les scénarios possibles.

Conclusion

Les MANETs se présentent comme des réseaux sans fil dans lesquels les équipements peuvent avoir des configurations différentes, et qui doivent coopérer pour assurer l'existence de tels réseaux et la communication entre les équipements du réseau. Les MANETs utilisent le lien radio. Ceci permet à un nœud malicieux d'interférer facilement pour perturber le fonctionnement du réseau.

Les protocoles de routage présentent des défis difficiles pour la sécurisation du routage, et la sécurité de routage dans ces réseaux est un prérequis pour leurs déploiements.

Il faut non seulement éviter de nombreuses attaques, mais aussi assurer la fiabilité des routages du réseau, car il existe plusieurs attaques qui ont comme but de surcharger le protocole, ce qui donne des effets néfastes sur le comportement du protocole.

Parmi ces attaques ; les attaques contre le protocole de routage AODV qui incluent attaque de largage de paquets, attaque par numéro de séquence, attaque de modification de champ ainsi attaque d'ajout de champ. Ce type d'attaques peut représenter une menace importante et dégrade facilement le bon fonctionnement du protocole.

Ce mémoire a été principalement axé sur une contribution à la sécurisation du protocole AODV, nous avons proposé une approche dans lequel on a sécurisé les paquets de routage en utilisant le protocole IPsec (AH) en mode transport et on a fait une simulation de cette approche.

Nous nous sommes intéressés à ses effets au niveau de routage, plus précisément le temps de découverte de routes. On a testé cette approche sur plusieurs scénarios en utilisant le simulateur OMNET++.

On a constaté une faible augmentation de temps de découverte de routes ce qui n'a pas de beaucoup d'effet sur les performances du réseau avec un gain important en termes de sécurité de routage.

Notre travail peut avoir des applications multiples dans différents domaines comme l'aviation, troupe de combats, réseaux de capteurs, etc...

Bibliographie

[01] Y. MELOUK, S. MOUHLI, Sécurité contre les attaques liées aux identités dans les réseaux Ad hoc, mémoire de fin d'études, Faculté des Sciences Exactes, Département d'Informatique, Université ABDERRAHMANE Mira Bejaïa, 2015/2016.

[02] F. AMEZA, Les technologies sans fil : Le routage dans les réseaux ad hoc (OLSR et AODV), Licence en informatique, Université de Bejaia, 2007

[03] N. Yahia, "Les réseaux ad hoc", publication : http://fr.calameo.com/books/0000007229c

[04] MABELE MONDONGA, Daniel. « Etude sur les protocoles de routage d'un réseau sans fil en mode Ad Hoc et leurs impacts ». Institut supérieur d'informatique, programmation et analyse de Kinshasa - Ingénieur informaticien 2010.

[05] S. Corsen and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Consideration (RFC 2501), Janvier 1999.

[06] AYAD Khadidja «Sécurité du routage dans les réseaux mobile Ad hoc », thèse de magistère, Ecole doctorale STIC à oued Smar ALGER, 2011 ,2012 .

[07] Wang Lan and Olariu Stephan, A two-zone hybrid routing protocol for mobile ad hoc networks, Parallel and Distributed Systems, IEEE Transactions on, Vol. 15, No 12, pp. 1105-1116, 2004

[08] A.BENALI Amal,M.DABO ,Routage réactif et proactif (Unicast) dans les réseaux mobiles Ad Hoc , Master en Informatique,Faculté des Sciences Exactes & Informatique,UNIVERSITE ABDELHAMID IBN BADIS MOSTAGANEM,2011/ 2012.

[09] E.M Royer and C-K Toh. "A review of current routing protocols for ad-hoc mobile wireless networks" . IEEE Personal Communications, Apr. 1999.

[10] C.E. Perldns and E.M. Royer. Ad Hoc On-demand Distance Vector Routing. Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps., Feb. 1999, pp. 90-100.

[11] Z.Haas et M. Pearlman, "The performance of query control schemes for the Zone Routing Protocol",IEEE selected area in communication , Août 1998.

[12] S. BENINE, le routage multi-chemin dans les réseaux de capteurs sans fil,mémoire de fin d'études Master, Système distribué & Intelligence Artificielle,Université Echahide Hamma Lakhdar El-oued,2014 – 2015.

[13] Houari M AOUCHI, Routage avec Qualité de Service dans AODV, mémoire Présenté pour obtenir le titre d'ingénieur d'état, Université Mouloud Mammeri de Tiz ouzou,2008-2009

[14] C. E. Perkins, E. E. Royer, S. R. Das, and M. K. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks, «In IEEE Personal Communications, Feb 2001, vol. 8, pp. 16–28.

[15] https://www.ietf.org/rfc/rfc3561.txt

[16] Yaser Yousef, Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil, Thèse de Doctorat Spécialité informatique, université de haute alsace, 2010

[17] BENINE Safa, „le routage multi-chemin dans les réseaux de capteurs sans fil". Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique , Université Echahide Hamma Lakhdar El-oued,2014-2015.

[18] MOHAMMED BELBACHIR , Stratégie de tolérance aux pannes pour un routage efficace dans les réseaux de capteurs ,Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique ,Université Abou Bakr Belkaid– Tlemcen,2014.

[19] http://rdoc.univ-sba.dz/bitstream/123456789/1925/3/D3C_Inf_AZZA_MOHAMMED.pdf

[20] https://dl.ummto.dz/bitstream/handle/ummto/6253/YahiSiham_MallekF.pdf

[21] BENAMAR KADRI, MOHAMMED FEHAM, ABDELLAH MHAMMED Secured and Optimized AODV for Wireless Sensor Networks, International Journal of Information Technology and Computer Science(IJITCS), Mai 2013.

[22] Benabdallah Karima, Optimisation d'un protocole de routage AODV dans les Réseaux de capteur sans fil, MASTER En Télécommunications, Réseaux Mobiles et Services de Télécommunications, Université Aboubakr Belkaïd – Tlemcen,2016/2017

[23] International Journal of Modern Engineering Research (IJMER) www.ijmer.com Vol.2, Issue.3, MayJune 2012 pp-728-732

[24] https://www.rfc-editor.org/rfc/pdfrfc/rfc3561.txt.pdf

[25] Y. Yahiatene, "Traffic Encryption Keys distribution models in Mobile Ad hoc Networks (Distribution de clés dans un réseau dynamique)", mémoire de magister de l'Université M'hammed Bougara, Boumerdes, 2011

[26] https://sg.inflibnet.ac.in/bitstream/10603/207580/12/12_chapter3.pdf

[27] NZALANKUMBU DIALEMBA, Etude des protocoles de sécurité dans le réseau internet , Institut supérieur de techniques appliquées Kinshasa , Ingénieur en informatique appliquée 2007.

[28] https://www.teal-consulting.de/en/2019/03/15/esae-series-part-5-windows-ipsec/

[29] https://wallu.pagesperso-orange.fr/pag-ipsec.htm

[30] https://www.frameip.com/vpn/

[31] https://db0nus869y26v.cloudfront.net/fr/IPsec

[32] https://www.ietf.org/rfc/rfc2402.txt

[33] https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.halz002/ipsecurity_ipsec_ah_esp.htm

[34] Alaedine BOUHAFS ,Layachi FENOUCHE , Mise en œuvre des VPNs: LAN-to-LAN et END-to-LAN en IPSec et IPSec/L2TP ,Université Abderrahmane MIRA-Bejaïa,2O11/2012.

[35] https://www.ietf.org/rfc/rfc2406.txt

[36] https://www.ccexpert.us/iscw/internet-key-exchange-ike.html

[37] Angelos D. Keromytis,Implementing Internet Key Exchange (IKE),Distributed Systems Lab, University of Pennsylvania,2000

[38] https://elearning-facsci.univ-naba.dz/pluginfile.php/35256/mod_resource/content/1/Chapitre%205-protocoles%20securis%C3%A9s.pdf

[39] BOUZELATA Hocine, 'Etude sur la conservation de l'énergie au niveau MAC des réseaux de capteurs sans fil (RCSF) et leur Simulation en utilisant le simulateur Castalia sur la plate forme OMNET++ '' ,2015.

[40] Mme HELAILI Nabila && Mme MEKHNACHE Salima.théme, promotion 2016-2017.

[41] http://www.omnetpp.org/

[42] OMNeT++, Installation Guide, Version 5.2, Copyright © 2014 András Varga and OpenSim Ltd.

[43] Berriha Saliha && Miloudi Takoia , 'Etude comparative de deux simulateurs pour les réseaux ad-hoc sans fil, promotion 2016-2018.

[44] András Varga: The OMNeT++ Discrete Event Simulation System. In the Proceedings of the European Simulation Multiconference (ESM'2001). June 6-9, 2001. Prague, Czech Republic.

[45] Mlle. Hania GATI,' Dissémination dans les Réseaux de Capteurs Véhiculaires, 2012.

[46] Dhafer ABIDI,'Etude et simulation du protocole TTethernet sur un sous-systhéme de gestion de vols et adaptation de la planification des taches a des fins de simulation,2015.

[47] Mme HELAILI Nabila && Mme MEKHNACHE Salima.théme, promotion 2016-2017.