UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH
FACULTÉ DES SCIENCES ET TECHNIQUES FÈS
DÉPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ETUDES

MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX

LES PROTOCOLES DE ROUTAGE GÉOGRAPHIQUE VANETS SÉCURISÉS

LIEU DE STAGE : LABORATOIRE SYSTÈMES INTELLIGENTS & APPLICATIONS

RÉALISÉ PAR : HAFIDHOU IBRAHIM AHMED SAID

SOUTENU LE 16 JUIN 2017

ENCADRÉ PAR :

MR. RACHID BEN ABBOU
MR. ABDELALI BOUSHABA

DEVANT LE JURY COMPOSÉ DE :

PR. RACHID BEN ABBOU
PR. ABDELALI BOUSHABA
PR. MOHAMED CHAOUKI ABOUNAIMA
PR. KHALID ZENKOUAR
PRE. ILHAM CHAKER

ANNÉE UNIVERSITAIRE 2016-2017

بِسْمِ اللهِ الرَّحْمَنِ الرَّحِيْمِ

رَبَّنَا عَلَيْكَ تَوَكَّلْنَا وإِلَيْكَ أَنَبْنَا وإِلَيْكَ المَصِيْرُ

رَبِّ اشْرَحْ لِي صَدْرِي وَيَسِّرْلِي أَمْرِي وَاحْلُلْ عُقْدَةً مِّن لِّسَانِي يَفْقَهُوا قَوْلِي

Dédicaces

Que ce travail témoigne de mes respects :

A mes parents,

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études. Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.
Je prie Allah de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.

A mes sœurs et à mes frères,

Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'ils n'ont cessé de me porter.

A mes encadrants et à tous mes professeurs,

Leur générosité et leur soutien m'obligent de leurs témoigner mon profond respect et ma loyale considération.

A mon ami Bouzite Bilal et à ma collègue Smiri Safae,

Vous trouverez ici le témoignage d'une fidélité et d'une amitié infinie.

A tous mes proches et à mes amis,

Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

Remerciements

Je souhaiterais avant tout présenter mes remerciements à **Allah** le tout puissant. C'est par votre volonté et votre miséricorde que j'ai pu réaliser ces travaux. Je vous remercie encore de m'avoir accordé la patience le long de mes études et je vous prie de m'en accorder encore dans le reste de ma vie.

Je tiens à exprimer mes remerciements les plus sincères au **Professeur Rachid BEN ABBOU** et au **Professeur Abdelali BOUSHABA**, mes directeurs de mémoire, pour m'avoir accepté de m'octroyer ce sujet de recherche et pour m'avoir guidé dans l'élaboration de ce travail, encouragé et soutenu tout au long de la mémoire avec patience et disponibilité, et pour la confiance qu'ils m'ont accordé. Je vous remercie du fond du cœur de vos encouragements permanents et je vous en suis très reconnaissant.

Je remercie très chaleureusement le **Professeur Azzedine ZAHI** d'avoir accepté de prendre part aux différentes réunions organisées en collaboration avec mes encadrants. Vos remarques me sont très chères et valent de l'or.

Je tiens également à remercier la **Professeure Ilham CHAKER** d'avoir accepté de présider cette soutenance. Qu'elle trouve ici le témoignage de ma gratitude.

Je dois aussi un grand merci au **Professeur Mohamed CHAOUKI ABOUNAIMA** et **au Professeur Khalid ZENKOUAR** de me faire l'honneur d'examiner ce travail. Qu'ils retrouvent ici le témoignage de ma profonde reconnaissance.

Merci également à ma collègue **Smiri SAFAE** et mon collègue **Bouzite BILAL** pour leurs aides et leur conseils toujours très pertinents.

J'adresse enfin mes plus tendres remerciements à mes chers **parents**, à mes **frères** et **sœurs**, à mes oncles et tantes, et à **toute ma famille.** Leur présence inconditionnelle a fait de moi la personne que je suis devenu.

Résumé

Les réseaux véhiculaires sont une nouvelle technologie réseau utilisée pour créer un réseau sans-fil entre des véhicules mobiles. Ils sont susceptibles de devenir la forme de réseaux mobiles Ad hoc la plus utilisée à l'avenir, bien que la sécurité dans ceux-ci représente un point essentiel qui est en cours d'élaboration.

Ce travail de recherche présente une étude sur l'importance de la sécurisation des protocoles de routage géographique VANETs, notamment le protocole GPSR (Greedy Perimeter Stateless Routing). Ce dernier se base sur les informations de localisation diffusées périodiquement à travers des messages échangés entre les véhicules (nœuds) pour découvrir les chemins de routage de la source à la destination.

Théoriquement, nous avons présenté quelques attaques pouvant être menées contre ce protocole de routage et une solution de sécurité se basant sur la génération de clés secrètes et la vérification des signatures numériques.
Nous nous sommes servis du simulateur réseau NS-2 pour simuler, évaluer et comparer les paramètres de performance de ce protocole sans attaques et lorsqu'on lui applique une attaque *Blackhole* dans un environnement VANET.

Mots clés : GPSR, protocole de routage, réseaux ad hoc véhiculaires (VANETs), routage géographique, sécurité de routage.

Abstract

Vehicular Ad Hoc NETworks (VANETs) are a new network technology used to create a wireless network between mobile vehicles. They are likely to become the most used form of Ad Hoc mobile networks in the future, although, in these, security represents an essential point which is still under development.

This research presents a study on the importance of the security of a VANETs geographic routing protocol, especially GPSR (Greedy Perimeter Stateless Routing). It is based on the location information periodically exchanged between vehicles to discover the routing path from the source to the destination.

Theoretically, we have presented some attacks that could be carried out against this routing protocol and security solution based on the generation and verification of digital signatures. We used the Network Simulator NS-2 to simulate, evaluate and compare the performance parameters of this Protocol without attack and when we apply a *Blackhole attack* in VANET environment.

Keywords: GPSR, routing protocol, Vehicular Ad Hoc Networks (VANETs), geographic routing, secure routing.

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

AODV	Ad-hoc On-demand Distance Vector
ARIADNE	A Secure On-Demand Routing Protocol for Ad Hoc Networks
AWK	Aho Weinberg Brian
CA	Central Authority
CANU	Communications in Ad Hoc Networks for Ubiquitous Computing
CBR	Constant Bit Rate
CORE	Collaborative REputation
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DoS	Denial of Service
DSR	Dynamic Source Routing
DSRC	Dedicated Short Range Communication
GF	Greedy Forwarding
GPS	Global Positioning System
GPSR	Greedy Perimeter State Routing
GSR	Geographic Source Routing
ID	Identity
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronic Engineers
IP	Internet Protocol
JVM	Java Virtual Machine
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork
NAM	Network AniMator
NS-2	Network Simulator 2
OBU	On-Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
OTCL	Object-oriented Tool Command Language
PDR	Packet Delivery Ratio
PF	Perimeter Forwarding

PKI	Public Key Infrastructure
QoS	Quality of Service
RREP	Route REPly
RREQ	Route REQuest
RSU	RoadSide Unit
SAODV	Secure Ad hoc On demand Distance Vector
SRP	Secure Routing Protocol
TCL	Tool Command Language
TCP	Transfer Control Protocol
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TTL	Time To Live
UDP	User Datagram Protocol
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
VANETs	Vehicular Ad hoc NETworks
VANATMOBISIM	Vehicular Ad hoc Network MOBIlity SIMulator
WAVE	Wireless Access of Vehicular Environment
XML	eXtensible Markup Language
ZBIDS	Zone Based Intrusion Detection System

Introduction générale

Aujourd'hui, les technologies sans fil connaissent un succès grandissant dans la société. La flexibilité et le développement rapide de ces technologies a fait d'elles un des domaines de recherche les plus attractifs. Les réseaux véhiculaires aussi appelés VANETs (Vehicular Ad hoc NETworks) [1] constituent une sous-classe des MANETs (Mobile Ad hoc NETworks) [2] et représentent une des composantes les plus prometteuses des Systèmes de Transport Intelligents (STI). Ces systèmes visent à intégrer les nouvelles technologies de l'information et de la communication dans le domaine des transports en vue d'améliorer la sécurité et le confort des usagers de la route. Dans les VANETs, les nœuds peuvent être des véhicules ou des infrastructures fixes appelées RSU (Road Side Unit) installées le long des routes. Les différents nœuds du réseau disposent d'équipements leur permettant de communiquer via des technologies sans fil.

Les protocoles de routage définissent la façon dont les différentes entités du réseau échangent des informations. Ceci comprend la phase d'établissement des routes, la décision de relayer les paquets et la phase de maintenance des routes. Le protocole de routage est donc crucial pour un déploiement harmonieux des applications dans un réseau. Vu l'importance des données échangées sur le réseau, un attaquant peut, en absence des mesures de sécurité, modifier le comportement des véhicules par l'ajout de fausses données dans le trafic, ou usurper une information liée à l'identité d'un véhicule dans les messages diffusés.

On note deux grandes familles de protocoles de routage dans les VANETs : les protocoles basés sur la topologie du réseau (AODV, OLSR, …) et les protocoles basés sur la position des nœuds (GSR, GPSR, …), dits aussi protocoles géographiques. Dans cette étude, nous nous intéressons à cette deuxième famille de protocoles et plus particulièrement au protocole GPSR (Greedy Perimeter Stateless Routing). Ce dernier utilise un mécanisme à deux étapes : la localisation du nœud destinataire et l'acheminement des paquets vers ce nœud.

Ce travail est divisé en quatre chapitres :

Le premier constitue une généralité sur les réseaux véhiculaires sans fil.

Dans le second chapitre, nous présentons des menaces et des solutions de sécurité de routage sur les VANETs.

Dans le troisième chapitre, nous allons présenter certains protocoles de routage géographique et particulièrement le protocole GPSR. Puis, nous étudierons certaines menaces et solutions de sécurité aux protocoles géographiques et plus spécifiquement le protocole GPSR.

Le dernier chapitre est consacré aux simulations et l'analyse des résultats. Dans ce chapitre nous présentons une étude comparative entre les deux protocoles topologique et géographique, respectivement AODV et GPSR sans attaque et dans le cas d'une attaque Blackhole.

Chapitre 1 - Généralité sur les VANETs

1.1 Introduction

Les réseaux véhiculaires sans fil dérivent de l'exploitation des technologies conçues pour les réseaux Ad hoc mobiles (MANETs). Leur élaboration accentue l'émergence des systèmes de transports intelligents (STI). Ces derniers ont pour but d'améliorer la sécurité routière et de rendre plus efficace le temps passé sur la route. Cela se fait par le biais des systèmes embarqués aussi bien dans les voitures qu'installés au bord des routes. Les conducteurs peuvent envoyer, ou recevoir des informations sur l'état des routes et des alertes sur des accidents de la route. Quant aux passagers des voitures, ils peuvent s'échanger des données (musique, vidéo) et d'autres informations utiles ; ce qui rend le temps passé sur la route, moins ennuyeux.

On distingue deux catégories de réseaux mobiles sans fil : les réseaux en mode infrastructure ou cellulaire nécessitant généralement l'installation des stations de base et les réseaux mobiles sans infrastructure ou Ad hoc se caractérisant par leur dynamisme et leur facilité de déploiement. Ces caractéristiques rendent ces derniers beaucoup plus utilisés dans diverses applications, comme la téléphonie, les applications militaires, les applications commerciales et la sécurité routière.

Dans ce chapitre, nous parlerons d'abord de la communication entre les différentes entités du réseau, et des services issus de ces réseaux. Ensuite, nous aborderons la norme et le standard de communication. Et enfin, nous présenterons les éléments et concepts de sécurité des réseaux VANETs.

1.2 Communication dans les réseaux VANETs

Dans les VANETs, les véhicules (entités mobiles) s'organisent pour établir la communication entre eux et aussi entre les entités fixes disposées le long de la route. L'échange des données entre les véhicules est désigné sous le nom de la communication en mode ad hoc alors que celui entre les véhicules et les entités fixes est connu sous l'application de communication en mode infrastructure. Dans cette section, nous détaillerons les principes et l'avantage de chaque mode de communication.

1.2.1 Communication en mode infrastructure

La communication en mode infrastructure est également connue sous le nom de la communication véhicule-infrastructure. Trois entités s'organisent pour établir ce type de communication :

- **OBU** : ensemble de composants logiciels embarqué dans le véhicule. Il permet aux véhicules de se localiser, de calculer et d'envoyer des données sur l'interface réseau.
- **RSU** (Road Side Unit) : entité installée au bord des routes, diffusant aux véhicules des informations sur l'état du trafic et sur les conditions météorologiques. Elle est utilisée comme point d'accès au réseau.
- **CA** (Central Authority) : C'est la Centrale d'autorité qui gère le réseau et qui joue le rôle de serveur de stockage des données. Elle délivre également des certificats et des clés ou pseudonymes de communication aux véhicules [3].

Le mode de communication infrastructure offre une meilleure connectivité et permet l'accès à divers services (Internet, informations météorologiques, …).
Toutefois, le déploiement des entités fixes le long des routes est très couteux. Aussi, la communication en mode infrastructure présente un temps de latence dans l'acheminement des paquets [4].

1.2.2 Communication en mode Ad hoc

Connue aussi au nom de communication véhicule à véhicule (V2V) [5], la communication Ad hoc dans les VANETs est similaire à celle des nœuds mobiles dans les MANETs. Dans ce mode de communication, chaque véhicule équipé d'une plateforme électronique appelée OBU (On-Board Unit), échange des données avec les véhicules situés dans sa zone de transmission radio. La communication Ad hoc fonctionne en environnement décentralisé et n'exige pas d'infrastructure pour son fonctionnement. Ce mode de communication est très efficace pour la diffusion rapide des informations liées aux services de sécurité routière.
Cependant, à cause de la forte mobilité des véhicules, la connectivité n'est pas permanente entre les véhicules. C'est ainsi que la combinaison des deux modes de communication dans les réseaux VANETs est souvent incontournable. La Figure 1 suivante décrit les modes de communication existants dans les réseaux véhiculaires sans fil.

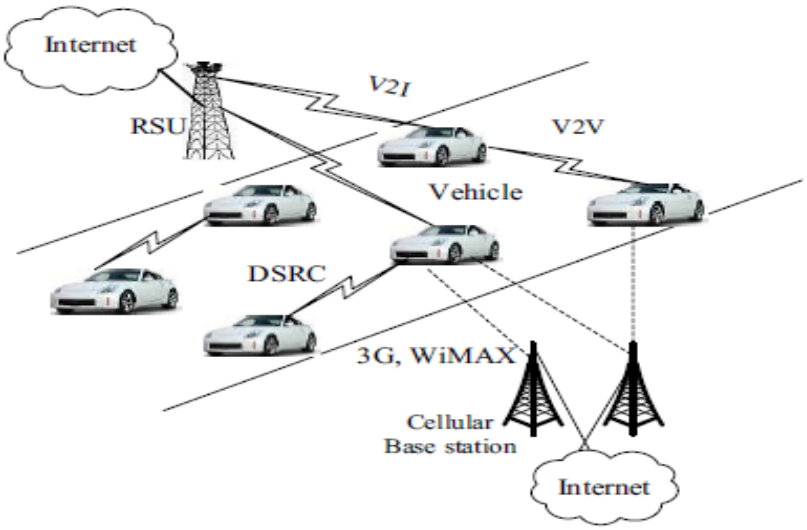


Figure 1: Communication dans les réseaux VANETs [6]

1.3 Services dans les réseaux VANETs

Les VANETs contribueront à la réduction du nombre d'accidents sur les routes grâce aux différents services que peuvent bénéficier ces utilisateurs. En effet, de la communication entre les différentes entités du réseau découlent trois types de services qui vont de la gestion du trafic routier à l'amélioration du confort des usagers [7].

1.3.1 Services de gestion et d'amélioration du trafic routier

L'amélioration du trafic routier se fait en utilisant les informations sur l'état des routes fournis par ces services. À partir du contenu des messages échangés par les différentes entités du réseau, un véhicule peut être informé de la circulation sur son trajet actuel ou futur. De ce fait, un conducteur peut par exemple décider de suivre une autre route lorsque le trafic est dense sur son trajet et éviter ainsi la congestion.
Ces services permettent aussi de créer le passage pour les voitures d'urgence (ambulance, voitures des pompiers, …), ou de proposer des itinéraires aux véhicules qui se dirigent dans une zone de congestion.

1.3.2 Services de prévention et de sécurité du trafic routier

Ces services jouent un rôle très important, dans la mesure où ils permettent aux conducteurs d'élargir leur champ de vision. En effet, par les messages d'alerte diffusés entre les différentes entités, les conducteurs peuvent être avertis des accidents ou autres situations dangereuses (informations météorologiques, alerte pour les travaux routiers) ayant un effet direct sur les personnes et les biens. Ces services contribuent à la diminution du nombre d'accidents sur les routes. Par conséquent, ils contribueront considérablement à la préservation de la vie humaine. Parmi ces services, on peut citer le service SOS qui est déjà implémenté dans certaines voitures actuelles. En cas d'accident, ce service envoie un message de prévention au centre de secours le plus proche. Ceci conduit à l'arrivée rapide de l'équipe de secours et ainsi prévenir un cambriolage [8].

1.3.3 Services d'amélioration du confort des usagers

En plus des services de prévention et de gestion du trafic routier, les VANETs contribuent également à l'amélioration du confort des usagers et leur permettent de bénéficier de plusieurs services, grâce à l'accès internet. À travers les réseaux véhiculaires sans fil, les conducteurs et les passagers des voitures peuvent recevoir instantanément des offres commerciales (des annonces des hôtels, des stations de services, des services d'informations touristiques, …), et des informations sur les lieux de stationnement dans leur zone de voisinage. De plus, par l'échange de données, les conducteurs et les passagers peuvent s'échanger des vidéos ou jouer en réseau. Dans ce type de réseau, il est possible de déployer la conduite assistée entre les conducteurs, la vérification à distance des permis de conduire et

des plaques d'immatriculation par les autorités compétentes, et le paiement électronique au niveau des points de péage afin de faire gagner du temps aux utilisateurs.

1.4 Norme et standard de communication

Pour mettre en place la communication entre les différentes entités dans les réseaux VANETs, l'ASTM (American Society for Testing and Materials) a adopté en 2002, une norme de communication appelée DSRC (Dedicated Short Range Communications) [9], dont la couche physique est basée sur la norme IEEE 802.11a [10]. En 2003, s'inspirant des travaux de l'ASTM, l'IEEE a étendu sa famille de standard 802.11 en y ajoutant le 802.11p [8], [10].
Dans cette section, nous allons présenter brièvement la norme DSRC et le standard 802.11p.

1.4.1 DSRC

Cette norme regroupe un ensemble de technologies dédiées aux communications véhiculaires. Cette technologie a évolué de la norme IEEE 802.11a à la norme 802.11p ou WAVE afin de répondre aux caractéristiques des réseaux véhiculaires VANETs.
Le DSRC propose un débit (atteignant 54 Mbit/s) suffisant pour le volume de données transporté. Avec une latence faible (inférieur à 5 ms), la technologie DSRC supporte une forte mobilité (aptitude à la mobilité élevée à 300 km/h) sur une portée maximale théorique de 1000 m, ainsi que le trafic de données temps réel. Un des avantages majeurs de cette technologie, c'est la capacité de s'adapter à tous les types de communication véhiculaires (V2I/V2V).

1.4.2 IEEE 802.11p

Ce standard utilise le concept de multicanaux dans le but d'assurer les communications liées aux applications de sécurité et autres services des transports intelligents [10], [11]. Il dérive de la couche physique du standard 802.11a, et il est aussi adapté au fonctionnement à faible charge du spectre DSRC [11]. Le standard IEEE 802.11p offre un débit compris entre 6 et 27 Mb/s sur une distance de 1000 m et avec une modulation de type OFDM (Orthogonal Frequency Division Multiplexing). La couche MAC de ce standard reprend le principe du CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) développé dans la couche MAC de l'IEEE 802.11e, afin de gérer la qualité de service (QoS) et le support du protocole de marquage de priorité (priorité d'accès) [10], [11].

1.5 Éléments et concepts de sécurité

Afin de respecter les concepts de sécurité, différents éléments sont pris en compte dans le développement des réseaux VANETs. Dans cette partie, nous allons présenter ces éléments et décrire les concepts de sécurité qui doivent être pris en compte dans la conception des protocoles de sécurité pour les réseaux véhiculaires sans fil.

1.5.1 Éléments de sécurité

a. Le Tamper-Proof Device (TPD)

C'est un dispositif qui, embarqué dans les voitures, permet de stocker les données confidentielles aux véhicules (clés privées, certificats). En plus de cela, le TPD se charge de signer les messages envoyés par les véhicules. Il est conçu de manière à détruire automatiquement toutes les informations stockées lors d'une manipulation matérielle grâce aux capteurs de nature diverse qu'il contient [12].

b. Les certificats dans les réseaux VANETs

Dans les réseaux VANETs, l'utilisation des certificats vise à renforcer les mesures de sécurité établies par les algorithmes cryptographiques. Il s'agit principalement des algorithmes de la cryptographie asymétrique. On distingue deux types de certificats [6] :

b.1 Le certificat à court terme

Il ne contient pas des données personnelles du conducteur. Ce type de certificat peut contenir un identifiant virtuel et des pseudonymes de communication permettant de garder l'anonymat du véhicule dans le réseau. Un véhicule ayant un profil anonyme dans le réseau n'est identifiable que par la centrale autorité. Généralement, le certificat à court terme est utilisé dans les protocoles de routage. Il faut souligner que chaque véhicule possède un seul certificat à long terme et plusieurs certificats à court terme.

b.2 Le certificat à long terme

Chaque véhicule a un certificat à long terme qui contient en plus des données qui lui sont propres, les informations personnelles sur son propriétaire. Ce certificat est utilisé pour le renouvellement des certificats à court terme, établir une communication avec l'autorité centrale, ou faire des demandes de pseudonymes.

1.5.2 Les requis de sécurité

Dans la conception des protocoles de sécurité des réseaux VANETs, les requis de sécurité tels que l'authentification, la confidentialité, l'intégrité des données, la non-répudiation, la disponibilité, la gestion de la vie privée et le contrôle d'accès, doivent être pris en compte.

a. L'authentification

L'authentification est l'un des principaux requis de sécurité de tout système. Pour les réseaux véhiculaires sans fil, il est très important de connaitre plusieurs informations sur le nœud émetteur tel que son identifiant, son adresse, ses propriétés, sa position géographique. L'authentification a pour objectif principal de contrôler les niveaux d'autorisation du véhicule dans le réseau. Dans les VANETs, l'authentification peut aider à la prévention des attaques dites *attaques Sybil* en spécifiant un identifiant unique pour chaque véhicule. Grâce à cette

technique, un véhicule ne pourra pas réclamer d'avoir plusieurs identifiants et de faire croire qu'il s'agit de plusieurs véhicules et ainsi perpétrer une attaque sur le réseau.
Plusieurs types d'authentifications ont été présenté dans [13] :

a.1 L'authentification de l'ID

C'est le fait qu'un nœud soit capable d'identifier les transmetteurs d'un message donné de façon unique. C'est par cette authentification que passe l'accès au réseau du véhicule émetteur.

a.2 L'authentification de la propriété

Elle aide à déterminer le type d'équipement qui est en communication. Il peut s'agir d'un autre véhicule, d'un « RSU » ou encore d'un autre équipement.

b. La confidentialité

Le principe de la confidentialité est de rendre l'information du réseau accessible uniquement aux entités qui se sont authentifiées dans le réseau. Elle protège donc les données du réseau contre toute écoute clandestine. On peut identifier deux niveaux de protection [8] :

- Le service global : il protège toutes les données transmises entre les utilisateurs du réseau pendant une période donnée ;
- Le service restreint : il assure la protection des messages par l'ajout de champs spécifiques à l'intérieur du message.

La confidentialité peut être mis en place en utilisant les clé public/privé pour le cryptage des messages durant la communication [14]. Généralement ce sont les algorithmes de cryptographie asymétrique et symétrique qui sont utilisés pour assurer le chiffrement et le déchiffrement des données.

c. L'intégrité des données

Le principe de l'intégrité repose sur deux concepts :

c.1 L'intégrité des messages

Fonction permettant de s'assurer que l'information envoyée par la source n'a pas subi d'altération avant d'arriver au destinataire ;

c.2 L'intégrité physique

Elle est liée aux matériels utilisés pour chiffrer et déchiffrer les messages. Cette fonction permet de s'assurer que le dispositif servant à l'envoi ou à la collecte d'information n'a pas subi de modification.
Le service d'intégrité des messages veille à ce que les messages diffusés entre les entités du réseau ne subissent aucune modification, duplication ou réorganisation.

Les mécanismes proactifs utilisés pour gérer l'intégrité des messages sont les fonctions de hachage et le MAC (Message Authentication Code).
L'intégrité physique est assurée par le TPD (Tamper-Proof Device) embarqué dans les véhicules et permettant d'assurer l'intégrité des messages.

c.3 La non-répudiation

Ce requis permet d'empêcher une entité de nier d'avoir participer à une communication. La non-répudiation permet donc au récepteur de prouver qu'il a reçu le message d'un tiers de communication. Ainsi, pour chaque message reçu, l'émetteur peut être clairement identifié [15].
Le but général de la non-répudiation est de collecter, de maintenir et de rendre disponibles toutes les évidences à propos d'un évènement ou d'une action, afin de résoudre des disputes à propos d'une occurrence ou non d'une action. La non-répudiation dépend donc de l'authentification. Le système peut ainsi identifier l'auteur d'un message malveillant [16]. La mise en place de la politique de non-répudiation dans les VANETs permet donc d'éliminer toute possibilité pour un attaquant d'injecter des données erronées sans être identifié.
Généralement, c'est la signature numérique qui est utilisée pour garantir la non-répudiation des messages des applications de sécurité et de gestion du trafic routier. Quant aux messages des applications de gestion de confort, la non-répudiation n'est pas aussi nécessaire sauf pour les messages impliquant des transactions financières.

c.4 La disponibilité

Le réseau et les applications doivent rester disponibles même en présence de panne dans le réseau. Ce requis permet non-seulement de sécuriser le système mais rend aussi celui-ci tolérant aux fautes. Ainsi les ressources doivent rester disponible jusqu'à ce que la panne soit réparée [17]. Un protocole de routage adéquat est nécessaire pour atteindre tous les récepteurs d'un message envoyé. Certains messages doivent rester circonscrits à un moment ou à un endroit défini, pour ne pas induire en erreur les véhicules si l'information n'est plus pertinente.
Plusieurs applications nécessitent une réponse rapide de la part des capteurs ou du réseau ad hoc, car le délai rendra le message obsolète. Ce qui peut générer des situations désastreuses. Il est donc primordial que les ressources soient disponibles en temps et lien opportuns.

c.5 La gestion de la vie privée

Les messages diffusés par les véhicules à travers le réseau véhiculaire sans fil peuvent leur être une source de menaces. Un attaquant pourra suivre un véhicule dans le réseau, recueillir toutes les données liées à ce véhicule ou à son propriétaire et les utiliser à des fins néfastes. Vu le danger que représente la traçabilité des données, il est important d'adopter des mesures de sécurité afin de gérer la vie privée dans les VANETs. L'utilisation d'un protocole de gestion de l'anonymat est l'un des moyens utilisés pour éviter la traçabilité illégale des véhicules et préserver par la suite la vie privée des conducteurs des véhicules.

c.6 Le contrôle d'accès

Il est important de contrôler les accès des entités aux ressources et services du réseau. Il faut définir dans un premier temps les nœuds qui peuvent se connecter au réseau et garantir par la même occasion que les utilisateurs se conforment aux politiques de sécurités mises en place; tel est l'objectif du service de contrôle d'accès. Certaines communications comme celle de la police ou d'autres autorités ne doivent pas être entendues par les autres usagers. L'accès à certains services fournit par les infrastructures est réservé à une catégorie d'usagers. Il est donc primordial de mettre en place un système qui permet de définir toutes ces politiques d'accès pour garantir le contrôle d'accès dans le réseau [18].

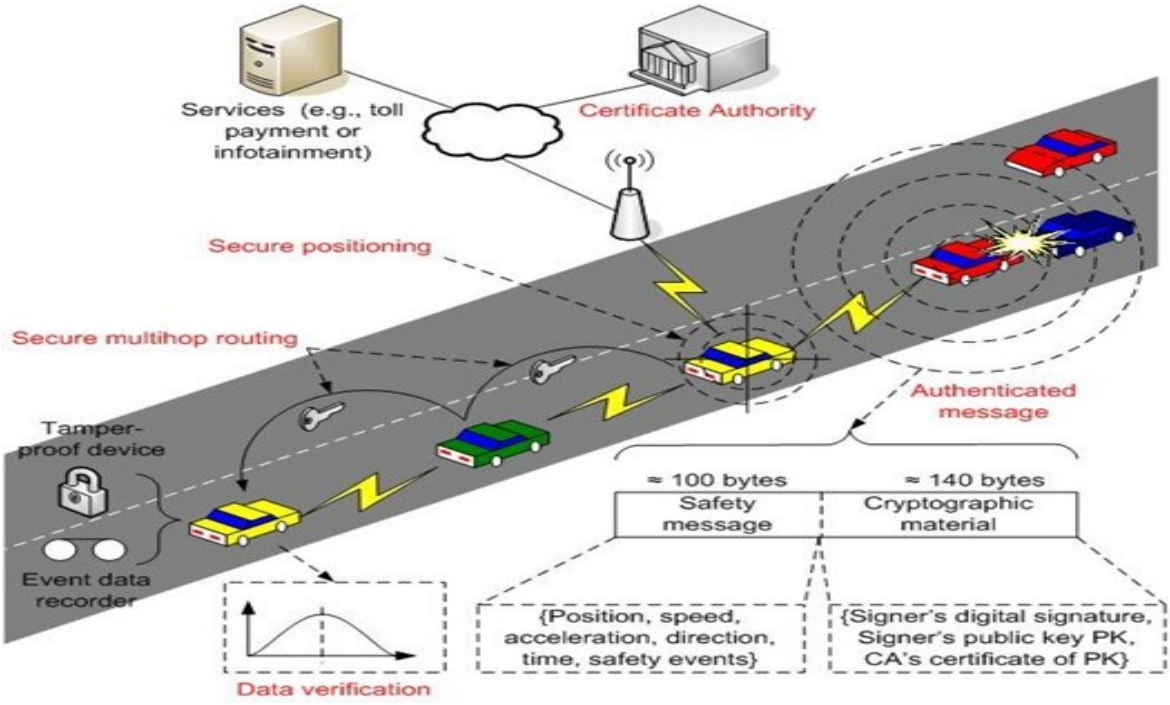


Figure 2: Éléments et concepts de sécurité dans les VANETs [19]

1.6 Conclusion

Nous avons présenté dans ce premier chapitre les réseaux véhiculaires sans fil et décrit les éléments et concepts de sécurité de ces réseaux. En raison des contraintes temps réel des applications des systèmes de transport intelligents et de la forte mobilité des nœuds, les protocoles de sécurité des réseaux filaires sont inadaptés pour les réseaux VANETs. Il faut donc définir de nouveaux mécanismes de sécurité afin de protéger la vie des différents utilisateurs des réseaux véhiculaires sans fil.

Chapitre 2 - Menaces et solutions de sécurisation pour les VANETs

2.1 Introduction

Comme dans chaque réseau classique, il existe plusieurs attaques de sécurité dans les VANETs. Le fait est que ces réseaux n'ont pas encore réellement été implémentés. Du coup, il est difficile de définir toutes les attaques pouvant y être perpétrés.
Les chercheurs en plus de considérer les attaques des MANETs, ont imaginé d'autres attaques dont pourrait être encouru les VANETs. Des solutions ont été proposées contre certaines attaques et d'autres sont en cours d'élaboration. Dans ce chapitre, nous verrons certaines attaques pouvant être menées contre les VANETs et certaines solutions de sécurité.

2.2 Menaces de sécurité dans les VANETs

Dans cette section, nous présentons certaines menaces que peut encourir un VANET. Cette classification est faite en considérant des paramètres tels que : l'étendu de l'attaque, l'impact de l'attaque sur le réseau, les requis à respecter pour ce réseau, les solutions possibles pour protéger le réseau et le profil des attaquants éventuels [14].

2.2.1 Le déni de service

Souvent dénoté par DoS (abréviation de l'expression en anglais « *Denial of Service* »), cette attaque est considérée comme la plus populaire dans les réseaux classiques et elle peut être aussi perpétrée dans les réseaux véhiculaires sans fil. Il consiste à rendre les différentes ressources et les services indisponibles pour les utilisateurs dans le réseau; il est généralement provoqué par d'autres attaques visant la bande passante ou les ressources énergétiques des autres nœuds. Un attaquant peut mettre en place cette attaque en inondant le réseau par des informations non pertinentes [19]. L'étendu de ce type d'attaques est généralement large, ce qui signifie qu'un grand nombre de nœuds peut être concerné. C'est donc une attaque qui peut s'étendre dans une zone géographique très large à travers plusieurs nœuds par des communications multi-sauts. De plus, l'impact de ce type d'attaques se reflète par le fait que l'attaque peut être détecté, mais il est difficile de la corriger [20], [21].

Dans une attaque de déni de service, certains messages pourraient être altérés si les lignes de communication ne sont pas disponibles. C'est donc le requis de l'intégrité des données qui est mise en cause par cette attaque. Il en est de même pour le requis de disponibilité, car si le réseau est indisponible, il y a une forte chance que les données ne soient pas disponibles pour les applications. Il est aussi évident que les contraintes de temps réels ne peuvent non plus être respectées dans ces conditions.

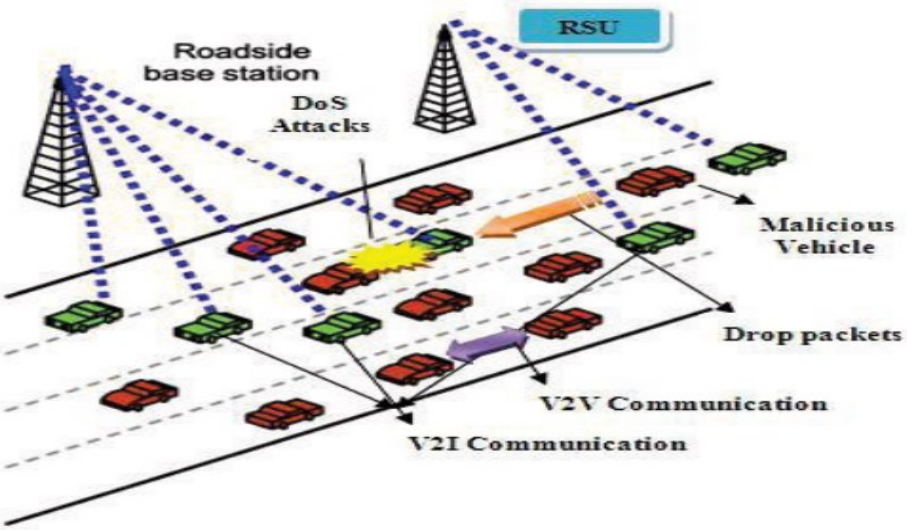


Figure 3: Attaque de déni de service [23]

2.2.2 L'écoute des messages

Dans ce type d'attaque, l'entité malveillante écoute sur le support de transmission afin d'extraire des informations sur le trafic échangé dans son voisinage. En effet, l'attaquant peut se positionner à une position dans un véhicule (en arrêt ou en mouvement) ou il peut aussi se présenter comme un faux « RSU » [22]. L'attaquant peut avoir comme objectif soit d'espionner sur des informations personnelles, ou bien de collecter des informations pour les analyser et effectuer ensuite d'autres types d'attaques.
Le requis de sécurité mis en cause dans ce type d'attaque est celui de confidentialité. Le cryptage des messages est l'une des solutions préconisées pour y faire face [22].

2.2.3 L'usurpation de l'identité d'un nœud

Ce type d'attaque consiste à prendre l'identité de quelqu'un d'autre et de faire croire que vous êtes cette personne [17], [22]. En effet, l'attaquant utilise l'identité d'un autre nœud afin de pouvoir recevoir ses messages ou des privilèges qui ne lui sont pas accordés. C'est une attaque qui fonctionne pour une communication à un saut car l'attaquant attaque directement sa cible sans passer par des nœuds intermédiaires [23]. Ce type d'attaque est difficile à détecter et même difficile à corriger, surtout si la cible est isolée. Les requis qui sont mis en cause dans ce type d'attaque sont : la non-répudiation, si l'identifiant est erroné, il est quasi-impossible de retrouver le nœud réellement fautif. La confidentialité et le contrôle d'accès sont aussi en cause dans ce type d'attaques. C'est par le fait que le nœud malicieux peut recevoir des informations en lieu et place du propriétaire de l'identifiant volé.
Un nœud malveillant peut par exemple usurper l'identité d'un autre nœud en utilisant des messages HELLO (*Identity spoofing*). Dans la Figure 4, le nœud malveillant *m* peut usurper l'identité du nœud *a* en envoyant des messages HELLO prétendant qu'il est le nœud *a*. Dans ce cas, les nœuds *i* et *g* vont annoncer à leurs voisins que le nœud *a* est accessible à travers le nœud *m*.

Ceci peut causer des conflits de routes vers le nœud *a* dans tout le réseau.

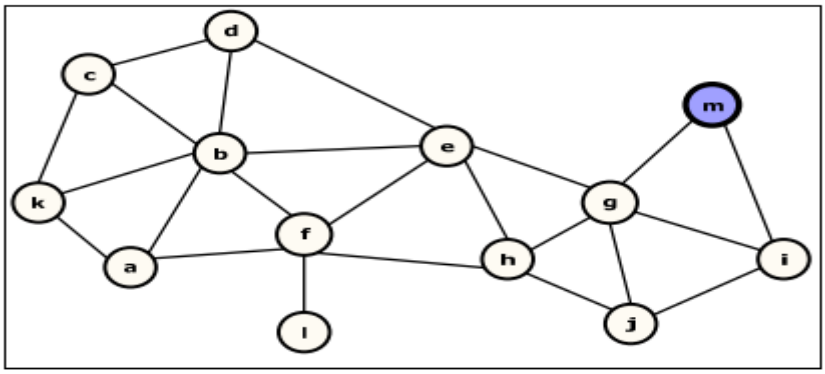


Figure 4: Usurpation d'identité du noeud a par m (HELLO) [26]

Les signatures numériques et le système de certificats permettent de prévenir ce type d'attaque [24].

2.2.4 L'injection des messages erronés

Dans ce type d'attaque, l'entité malveillante crée des messages contenant des informations erronées afin de causer un accident ou de rediriger le trafic routier de manière permettant de libérer la route utilisée. L'attaquant peut aussi changer le contenu ou le type d'un message lors de son passage à travers un nœud. Ce type d'attaque est souvent perpétré par un nœud intermédiaire par lequel le message transite pour retrouver son récepteur. C'est donc une attaque qui est perpétré lors de communications multi-sauts. Cette attaque peut être détectée si le message transite par d'autres nœuds dans le réseau. Dans ce cas, il est possible de déterminer que l'information provenant de ce nœud est différent de celle provenant d'autres nœuds du réseau. Mais si le nœud adversaire est le seul par lequel le message transite, il sera difficile de détecter et d'éviter l'attaque [25]. Dans ce type d'attaque, le requis concerné est l'intégrité, car le message étant altéré, son intégrité n'est plus garantie. La vérification des messages peut être utilisée pour prévenir ce type d'attaque [26].

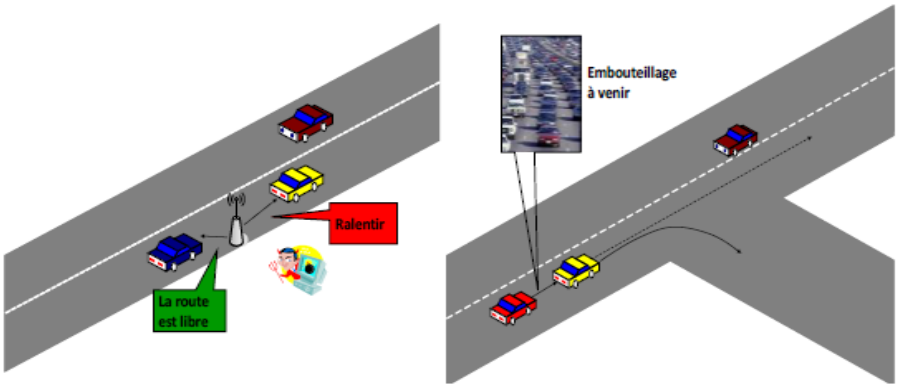


Figure 5: Attaques par l'envoi de messages falsifiés [30]

2.2.5 La révélation d'identité et de position géographique des autres véhicules

Dans ce type d'attaque, l'entité malveillante collecte des informations sur les transmissions radio effectuées par le véhicule victime afin de surveiller sa trajectoire. L'utilité de cette attaque est diverse et dépend de l'entité collectant ces informations. Cette entité peut être par exemple une entreprise de location de voitures qui veut suivre ses propres véhicules de manière illégitime.

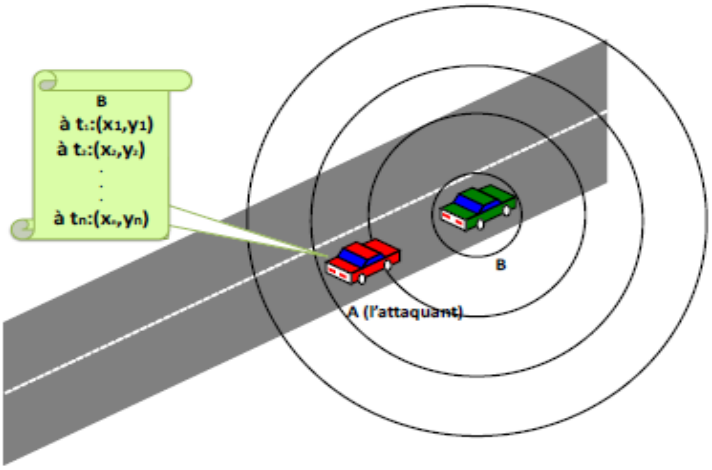


Figure 6: Attaque de révélation d'identité et de position géographique [31]

2.2.6 L'attaque Whormhole

Cette attaque consiste à rediriger le trafic entre deux zones géographiques éloignées pour ainsi avoir une bonne position géographique afin de contrôler le trafic qui passe par lui.

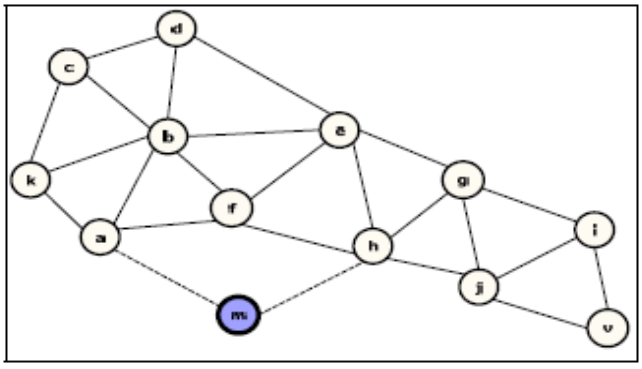


Figure 7: Attaque wormhole créée par le noeud m [26]

Dans la Figure 7, le nœud malicieux *m* crée un lien virtuel entre les nœuds *a* et *h* sans être visible par les deux nœuds. Le but est de leur faire croire qu'ils sont des nœuds voisins.

En effet, le nœud *m* renvoie les messages HELLO (messages de contrôle) du nœud *h* vers *a* et inversement. Ainsi, chacun de ces nœuds va déclarer par la suite qu'il a un lien symétrique entre eux. La route entre a et h devient donc une route préférée par les autres nœuds car c'est le plus court chemin. Étant totalement contrôlé par le nœud malveillant *m*, ce chemin présente un danger pour l'intégrité et la confidentialité des messages.
Plusieurs nœuds malicieux peuvent collaborer pour réaliser une attaque *wormhole*. Dans la Figure 8, les deux nœuds malicieux *m1* et *m2* collaborent pour créer une attaque *wormhole* entre deux zones très éloignées.

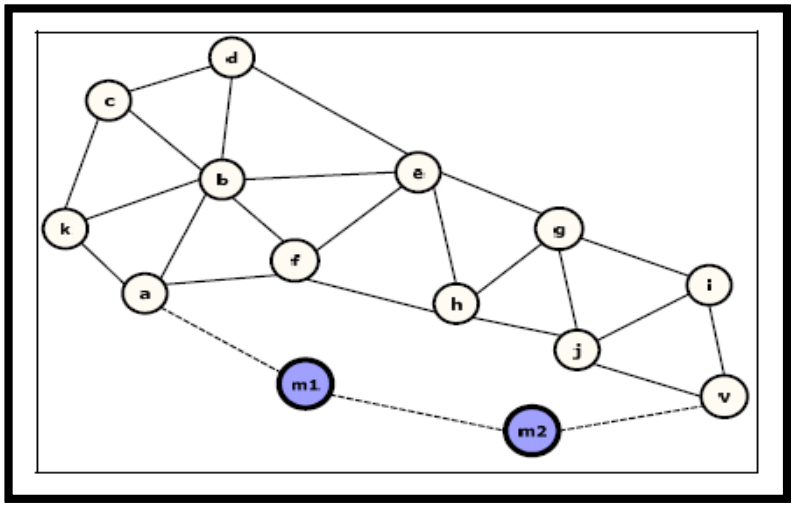


Figure 8: Collaboration pour créer une attaque wormhole [26]

2.2.7 L'attaque trou-noir (*en anglais, Blackhole Attack*)

Un nœud malicieux qui a été choisi par ses voisins comme relayeur peut rejeter tous les paquets de données reçus de ses voisins. Ce type d'attaque entraine une perte de connectivité et la dégradation de la communication.

Nous parlerons en détails de cette attaque dans le chapitre suivant, car celle-ci est appliquée au protocole GPSR.

2.3 Solutions de sécurité dans les VANETs

La sécurité est un sujet important à traiter, surtout pour les applications des réseaux VANETs qui sont très sensibles à la sécurité. Nous avons vu dans la première partie de ce chapitre qu'un attaquant peut émettre des messages d'alerte dont le contenu est falsifié ou empêcher l'acheminement d'un message légitime afin de causer des accidents. Pour ce faire, l'attaquant peut viser la disponibilité du réseau aux niveaux des différentes couches de la pile protocolaire. Le routage ad hoc est très différent de celui des réseaux traditionnels. C'est aussi le cas pour sa sécurité. Concrètement, pour sécuriser le routage dans un réseau traditionnel, il est suffisant de sécuriser des routeurs dédiés en les authentifiant par exemple.

Mais pour assurer la sécurité du routage dans un réseau VANET, chaque nœud doit non seulement prendre la responsabilité de ses propres comportements mais aussi de vérifier les comportements des autres nœuds.

2.3.1 Sécurité des VANETs *vs* des Réseaux Traditionnels

Nous nous focalisons sur la sécurité du routage ad hoc véhiculaire. Nous allons dans un premier temps discuter les raisons pour lesquelles les mécanismes de sécurité conçus pour les réseaux traditionnels ne sont pas adaptés aux réseaux ad hoc, puis parler en deuxième lieu des nouveaux besoins de sécurité du routage ad hoc véhiculaire.

Raison 1 : la mobilité

La mobilité des nœuds rend la topologie des VANETs instable. Il n'est donc pas facile pour un nœud de connaître correctement son voisinage. Les attaquants peuvent aussi forger et diffuser des fausses informations de topologie pour construire des routes qui passent par eux et réaliser ainsi des attaques qui visent à causer des accidents ou la congestion de routes. Par ce moyen, un protocole de routage ad hoc non-sécurisé peut facilement être attaqué. La mobilité des attaquants peut aussi les rendre plus difficiles à détecter ou à localiser.

En comparaison avec les réseaux traditionnels, il n'y a pas autant de mobilité dans les réseaux filaires, et dans les réseaux cellulaires ce sont des infrastructures qui gèrent la mobilité. Il est donc nécessaire de construire des protocoles de routage spécialement pour les VANETs. Et ces protocoles doivent être capables de découvrir correctement la topologie du réseau même en cas d'attaques.

Raison 2 : le support sans fil partagé

La nature de transmission radio dans l'air permet à un intrus d'écouter passivement [8] tous les messages échangés pourvu qu'il se trouve dans la zone d'émission. Il suffit qu'il opère en « promiscous mode » et qu'il utilise un logiciel qui permet de capturer les paquets émis (*sniffer*). L'adversaire aura donc accès au réseau et peut intercepter aisément les données transmises, sans même que l'émetteur ait connaissance de l'intrusion. Etant potentiellement invisible, l'intrus peut brouiller le canal radio pour bloquer les transmissions, injecter massivement des paquets visant à épuiser les ressources des nœuds, enregistrer, modifier, et ensuite retransmettre les paquets comme s'ils avaient été envoyés par un utilisateur légitime.

Pour éviter tout cela, les VANETs ont besoin de nouveau mécanismes afin de sécuriser l'accès au réseau.

Raison 3 : manque de serveurs centraux

Puisqu'il n'y a pas forcément de serveur central [27] dans les VANETs, la distribution et la gestion de clés peuvent être difficile à réaliser. Dans le cas des réseaux traditionnels, les solutions de sécurité se basent souvent sur des relations de confiance préalablement établies ou des autorités de confiance tierces. Des primitives cryptographiques sont aussi utilisées pour authentifier les nœuds et sécuriser les échanges de données.

Pour pouvoir utiliser ces moyens cryptographiques dans les VANETs, il faut étudier comment établir des autorités de confiance ou des relations de confiance entre les nœuds sans l'aide d'une infrastructure.

Raison 4 : manque de coopération

Dans un réseau ad hoc, il est souvent difficile de détecter des nœuds égoïstes [27] qui peuvent tout simplement être silencieux et/ou refusent de transférer les données afin de préserver leur ressource. Quand de tels nœuds sont nombreux dans le réseau, la disponibilité du service de routage sera mise en cause. Ce problème d'égoïsme n'existe pas dans les réseaux traditionnels où les nœuds reposent sur les routeurs dédiés pour assurer la fonctionnalité de routage.

Dans les réseaux VANETs, de nouveaux mécanismes doivent être désignés pour garantir la coopération des nœuds.

Raison 5 : nœuds compromis

En comparaison aux nœuds des réseaux traditionnels, ceux des réseaux VANETs sont plus faciles à compromettre [28], parce qu'ils sont de nature mobile. Par le fait que les VANETs peuvent être divisés et/ou fusionnés, les attaquants auront plus de chances d'attaquer des nœuds sans être aperçus. Comme nous l'avons vu précédemment, les attaques de type « wormhole » sont des attaques très sophistiquées ne pouvant être commises que par des nœuds compromis et sont difficiles à éviter.

L'utilisation de la cryptographie ne permet pas de résoudre le problème de ces nœuds compromis par une simple authentification. Ceci s'explique par le fait que ces nœuds ont été des participants légitimes au processus de routage. C'est pourquoi d'autres solutions doivent être envisagées pour palier à ce problème.

Après avoir eu une vue globale des nouveaux besoins de sécurité des VANETs, nous allons présenter dans la suite de ce chapitre, certaines solutions proposées pour la sécurité du routage ad hoc véhiculaire dans différents travaux de recherches [23], [29].

2.3.2 Solutions de sécurité de routage VANET

Pour faire face aux attaques décrites dans la première partie de ce chapitre, de nombreux mécanismes de sécurité de routage ont été proposés dans la littérature qu'on peut les classer en trois catégories, dont nous allons les détailler par la suite.

a. Mécanismes de gestion de clés

Ces mécanismes traitent l'identification et toutes les questions liées à la création, la distribution, la révocation, le renouvellement et l'échange des clés. Un système de distribution de clés dans les réseaux VANETs peut être asymétrique ou symétrique. Dans le cas des systèmes asymétriques, chaque nœud possède une paire de clés publique/privé gérée par une PKI (Public Key Infrastructure).

Quant aux systèmes symétriques, soit une clé symétrique est partagée par tous nœuds du réseau, ou plusieurs paires de clés symétriques partagées par chaque deux ou plusieurs nœuds.

a.1 Gestion de clés asymétriques

Le déploiement des PKI traditionnelles dans les réseaux VANETs est problématique. En effet, un tel système a besoin d'une autorité de certification (CA) qui est un serveur central qui assure la livraison et la révocation de certificats en permanence. En plus de cela, le CA doit être toujours connecté et accessible par les nœuds. Ces contraintes font que le PKI traditionnelle inadaptée à un environnement VANET.

Pour faire face à ces contraintes, Capkun et Hubeau [30] ont proposé une infrastructure à *clé public auto-organisée* inspirée du cryptosystème PGP (Pretty Good Privacy) pour authentifier les nœuds d'un réseau mobile. Dans cette PKI, chaque nœud établi des certificats pour les nœuds en qui il a confiance. Et si deux nœuds veulent communiquer sans qu'ils se connaissent préalablement, ils s'échangent leur liste de certificat afin de créer un certificat en commun.

Par exemple, si un nœud A veut communiquer avec un nœud C, et que le nœud A fait confiance en un troisième nœud B, alors A peut établir une chaine de confiance à travers B.

a.2 Gestion de clés symétriques

L'échange de clés symétriques a pour but d'établir une clé secrète commune entre les parties communicantes sans avoir aucune information préalable l'une sur l'autre. Parmi les protocoles d'échanges de clés, on peut citer celui inventé par Diffie et Hellman.

L'échange de clés Diffie-Hellman [31], du nom de ses auteurs *Whitfield Diffie* et *Martin Hellman*, est une méthode par laquelle deux nœuds Nœud 1 et Nœud 2 peuvent se mettre d'accord sur une clé qu'ils peuvent utiliser pour chiffrer une conversation.
Le protocole d'échange de clés de Diffie-Hellman, repose sur une fonction de la forme :

$K = g^x \bmod P$, avec P premier et g < P.

Une telle fonction est très facile à calculer, mais la connaissance de K ne permet pas d'en déduire facilement x. Cette fonction est publique, ainsi que les valeurs de g et P.

L'échange Diffie-Hellman se déroule comme suit (dans les calculs, g est utilisé comme générateur) :

1. Le Nœud 1 tire au hasard un entier a, tel que : $1 < a < P - 1$ et le garde secret.
2. Le Nœud 1 envoie à Nœud 2 : $A = g^a \bmod P$
3. Le Nœud 2 choisit un nombre b, tel que : $1 < b < P - 1$ et le garde secret.
4. Le Nœud 2 envoie à Nœud 1 : $B = g^b \bmod P$

5. Le Nœud 1 a reçu *B* et calcul $B^a \bmod P$ (en passant par, $(g^b)^a \bmod P$, mais il ne connaît pas B) : $S = B^a \bmod P$

6. Le Nœud 2 a reçu A et calcul $A = g^a \bmod P$ (en passant par, $(g^a)^b \bmod P$, mais il ne connaît pas A) : $S = B^a \bmod P$

Les Nœud 1 et Nœud 2 obtiennent à la fin de leurs calculs respectifs le même nombre qui n'a jamais été exposé à la vue des indiscrets : c'est la clé S.
La Figure 9 suivante présente le processus d'échange de clé Diffie-Hellman.

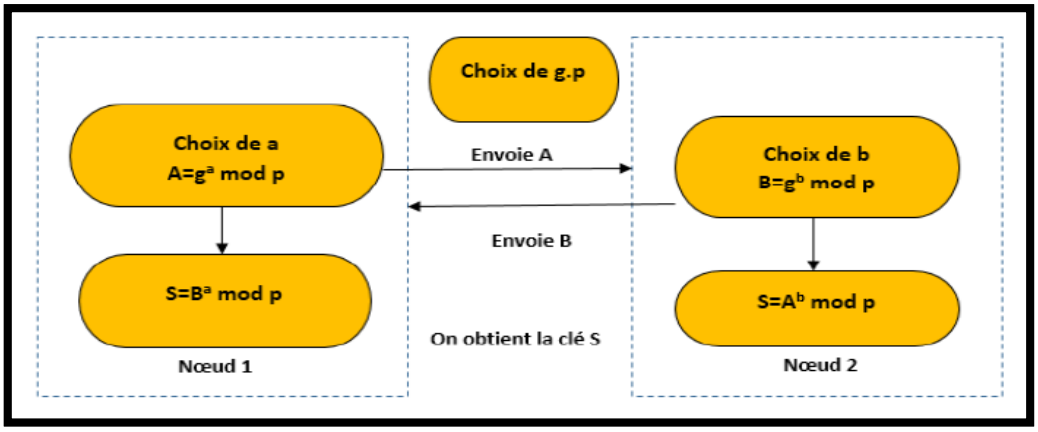


Figure 9: Echange de clé Deffie-Hellman [37]

De l'étude précédente, nous constatons que la solution symétrique semble être la plus adaptée pour les VANETs pour sa facilité de déploiement et sa rapidité de calcul.

b. Mécanismes de routage sécurisé

Ce sont des mécanismes garantissant l'authentification, la confidentialité, l'intégrité et finalement la non-répudiation dans les deux phases du service de routage : découverte de route et la transmission des données. Dans la littérature, des protocoles de routage sécurisés qui sont en effet des protocoles de routages existants renforcés par ces mécanismes de sécurité supplémentaires.
Nous présentons ci-dessous certains de ces protocoles sécurisés.

b.1 ARIADNE

C'est un protocole proposé par Hu et al. [32], qui est une extension au protocole DSR se basant sur la cryptographie symétrique utilisée par la technique d'authentification TESLA (Timed Efficient Stream Loss-tolerant Authentication).

Dans cette version sécurisée, un émetteur ajoute un (*HMAC*) avec des clés générées selon des intervalles de temps. Le récepteur peut vérifier le message lorsque la clé est envoyée dans un intervalle de temps futur sur la base d'un intervalle de divulgation de clé. La divulgation de clé temporisée est fondée sur le détachement, mais la synchronisation d'horloge est bornée entre les deux nœuds impliqués.

L'objectif principal d'ARIADNE est de fournir l'authentification et l'intégrité des messages de signalisation DSR. Dans la découverte de route, à chaque saut les nouvelles informations du RREQ sont authentifiées. L'état de la sécurité TESLA est vérifiée à la destination, et un HMAC est inclue dans la réponse de route RREP pour certifier que les conditions de sécurité ont été vérifiées.

Selon ses auteurs, le protocole peut également être adapté à deux autres schémas, dont les clés partagées entre chaque paire de nœuds et la signature numérique. L'inconvénient de la première variante est l'utilisation de TESLA qu'elle procure des délais supplémentaires dans le processus de découverte de route ; par conséquent, étant basé sur TESLA, ARIADNE n'est pas recommandé pour les réseaux à forte mobilité comme les VANETs.

Quant à l'autre variante ARIADNE basée sur la signature numérique, elle n'est pas adaptée pour un environnement véhiculaire à cause de la difficulté de distribution des clés publics qui nécessite un serveur central.

b.2 SRP

SRP (Secure Routing Protocol) [33] est aussi une extension sécurisée au protocole de routage réactif DSR se basant sur la cryptographie symétrique. SRP ne suppose pas que tous les nœuds intermédiaires dans une route, partagent une clé secrète avec le nœud source ou destinataire. Par conséquent, seuls les deux nœuds communicants (nœud source et nœud destination) partagent une clé secrète. De ceci résulte que la vérification et l'authentification de paquets de contrôle échangés ne sont effectuées qu'au niveau des nœuds communicants.

Au début, un nœud source S diffuse à ses voisins à un saut un paquet RREQ contenant un MAC calculé sur les différents champs avec la clé qu'il partage avec le nœud destinataire D. Ensuite chaque nœud intermédiaire recevant ce paquet ajoute son identifiant au descriptif de ce dernier. Dès que le nœud destinataire D intercepte le paquet contenant la route spécifiée dans le descriptif du paquet, il vérifie le MAC généré par S et ensuite envoie un paquet RREP à S via la route inverse; ce paquet contient le descriptif de la route trouvée et un MAC (calculé avec la clé secrète partagée avec S).

Les opérations du protocole SRP sont très optimisées en termes de bande passante et de traitement. Cependant, cet avantage disparait en présence d'un attaquant qui peut ajouter des paquets RREQ falsifiés (par exemple contenant un identifiant d'un destinataire inexistant) ou altérer le descriptif d'une route en cours de construction. Ainsi, l'authentification et la vérification d'intégrité de ces paquets ne sont pas effectuées par les nœuds intermédiaires. Ceci peut provoquer une consommation de ressources supplémentaires en présence d'attaquants.

b.3 SAODV

SAODV (Secure Ad hoc On demand Distance Vector) [34] est une extension du protocole AODV conçu pour assurer l'authenticité et l'intégrité des messages de routage, et pour éviter les manipulations néfastes de la valeur de nombre de sauts (HOP-COUNT).

Le SAODV agit sur les deux parties modifiable et non modifiable des messages de routage (Route_Reply et Route_Request) du protocole AODV. La partie modifiable qui inclut le compteur de sauts est protégée par une technique basée sur les chaînes de hachage, qui permet aux nœuds intermédiaires (selon les concepteurs de ce protocole) de vérifier que sa

valeur n'a pas été décrémentée d'une manière abusive. Tandis que la partie non modifiable est protégée par une signature numérique et elle inclut les champs suivants : le numéro de séquence, les adresses des nœuds source et destination et l'identifiant de requête.

La première technique de protection n'est toutefois pas totalement sûre et la valeur de compteur du nombre de sauts peut être rendue supérieure ou inférieure à sa valeur réelle par un nœud malveillant. Donc, le nœud malveillant peut faire apparaitre les routes plus courtes. Ce protocole présente aussi l'inconvénient que les nœuds doivent utiliser un serveur en ligne afin de vérifier les signatures numériques.

La Figure 10 illustre les techniques cryptographiques utilisées dans les trois protocoles sécurisés cités précédemment.

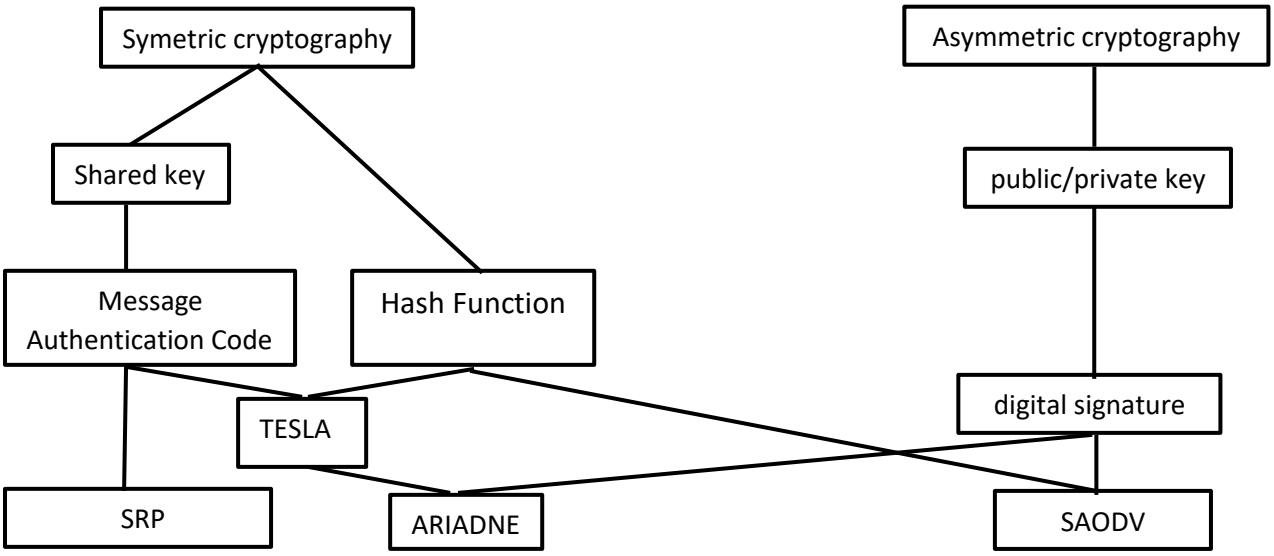


Figure 10: Protocoles de routage sécurisés

c. Systèmes de détection d'intrusions

L'utilisation des techniques cryptographiques n'offre pas la possibilité de détecter de nouvelles attaques, ni même de défendre le réseau contre des nœuds internes compromis [28].

Toutefois, ce type de système est utilisé comme première ligne de défense alors que les systèmes de détection d'intrusion occupent la deuxième ligne. Communément désignés par son acronyme anglais IDS (Intrusion Detection System), un IDS fonctionne en trois phases : une phase de collection de données suivie d'une phase d'analyse et enfin une phase de réponse pour prévenir ou minimiser l'impact sur le système. Le système IDS est implanté au niveau de certains nœuds spéciaux appelés moniteurs ou nœuds de surveillance. Le déploiement de ces nœuds diffère en fonction du type protocole et de l'architecture de l'IDS. Les IDS peuvent être classifiés [35] selon les techniques de détection utilisées :

• **système de détection d'anomalie** : il sert à détecter tout comportement qui dévie le comportement normal préétabli et déclenche une réponse.

• **système basé sur les signatures** : il possède une base de données de certaines attaques avec laquelle sont comparées les données collectées. Une attaque est détectée si les données collectées coïncident avec un comportement malicieux déjà enregistré dans la base de données.

• **système basé sur les spécifications** : ce système permet de définir un ensemble de conditions qu'un protocole doit satisfaire. Une attaque est détectée si le programme ou le protocole ne respecte pas les conditions de bon fonctionnement établies par le système.

Les IDS peuvent aussi être classés selon l'architecture en : autonome, distribuée et coopérative et hiérarchique [36].

c.1 Watchdog and Pathrater

Marti, Giuli, and Baker [37] ont présenté une solution pour détecter les nœuds malicieux qui suppriment les paquets (de façon sélective ou aléatoire) passant par ce nœud de transit. Cette solution nommé Watchdog consiste d'une part, à surveiller le comportement de tous les nœuds, et d'autre part, à choisir la route la plus sécuritaire grâce au module nommé Pathrater. De ce fait, tous les nœuds du réseau se surveillent les uns les autres sous forme d'une architecture maillée.

En effet, si un nœud source S veut transmettre un paquet vers un nœud destinataire D via les nœuds intermédiaires A, B et C, le paquet est transmis au nœud A qui le retransmet à son tour au nœud B mais garde une copie du paquet. La prochaine étape du processus est de surveiller si B va retransmettre ce paquet vers le nœud C en écoutant et en comparant tous les paquets émis par le nœud B. Si le nœud B ne retransmet pas le paquet au bout d'un certain temps, un compteur est incrémenté. Si le compteur atteint une valeur maximale préétablie (le nombre de fois que le nœud B ne transmet pas un paquet), le nœud A peut conclure que le nœud B est malicieux. Sa décision sera donc rapportée au nœud S.
Le mécanisme Watchdog est présenté dans la Figure 11 suivante :

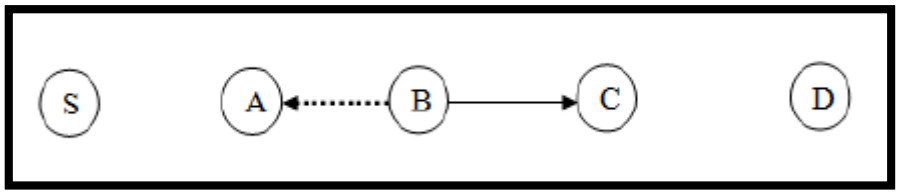


Figure 11: Principe du Watchdog [37]

Ces deux techniques sont très efficaces pour le choix des routes, car elles permettent d'éviter les nœuds malveillants. Toutefois, ces nœuds ne sont pas définitivement éliminés du réseau. En revanche, ils continuent à bénéficier du réseau puisque les autres nœuds transmettent les paquets vers ceux-ci, alors qu'ils suppriment les paquets afin d'économiser leurs propres ressources. Par conséquent, les nœuds malveillants sont encouragés à poursuivre leurs comportements.

c.2 CORE : système basé sur la réputation

CORE [38] est une solution proposée par *Michiardi* et *Molva* pour contrer le comportement égoïste des nœuds. La solution consiste à offrir des moyens d'incitation à tout nœud voulant participer aux processus collaboratifs. Les moyens d'incitation sont inspirés de la théorie des jeux (Prisoner's Dilemma) ou chaque nœud a une réputation à mettre en jeu traduisant son honnêteté. En effet, pour transmettre ou recevoir un paquet, le noeud doit avoir une réputation suffisante. De plus, tout noeud malveillant ou égoïste détecté verra sa réputation diminuer. Ceci a pour conséquence d'isoler ce noeud complètement du réseau (impossibilité d'émettre ou recevoir des paquets). Cela est très bénéfique, car tous les nœuds seront obligés d'adopter un comportement honnête.

Dans CORE, les nœuds impliqués dans un processus coopératif s'attribuent mutuellement une valeur de réputation. À noter que CORE ne permet d'attribuer que des valeurs positives pour la réputation, si le noeud de décision reçoit un rapport positif d'un autre noeud (surveillance indirecte). Les valeurs négatives sont réservées seulement pour une surveillance directe dans le cas où le noeud surveillé ne coopère pas.

En procédant ainsi, le mécanisme élimine les éventuelles fausses accusations et les attaques de dénis de service. Si un noeud A sollicite un service d'un noeud B (Retransmission de paquet ou découverte de routes), le noeud B consulte sa table de réputation et calcule la valeur globale de réputation (surveillance directe et indirecte) pour le noeud A. S'il s'avère que le noeud A à une réputation globale négative alors sa requête sera rejetée et le noeud sera isolé.

c.3 IDS de Zhang et Lee

C'est un IDS distribué et coopératif proposé par *Zhang* et *Lee* [39] où chaque noeud appelé agent IDS, est responsable de la collection des données et la détection des activités malicieuses.

Chaque agent IDS peut initier une réponse (punition) indépendamment des autres nœuds. Toutefois, les agents IDS voisins pourraient coopérer entre eux pour une détection d'intrusion globale.

Un agent IDS est structuré en six modules comme illustrés sur la Figure 12 :

- le module **local data collection** qui est responsable de la collecte des données en temps réel.
- le module **local detection engine** décide, à partir des données collectées, si le système est attaqué ou non. Le module peut initier une réponse si une attaque est détectée.

La réponse est exécutée par le module **local response** (alerte à l'utilisateur local) ou le module **global response** (alerte globale) en fonction du type d'attaque.

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

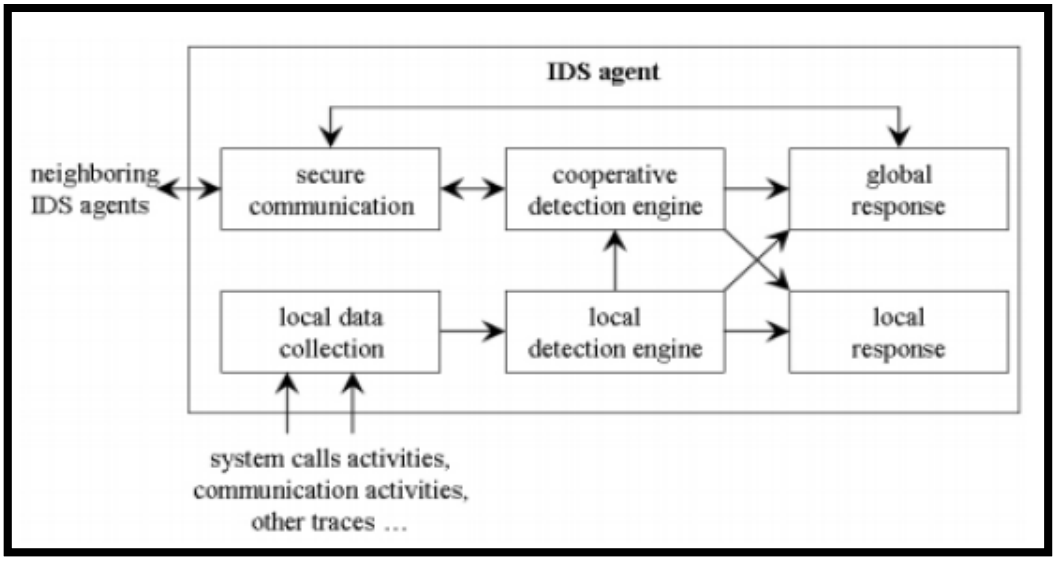


Figure 12: Modèle d'un agent IDS [39]

c.4 Zone-Based Intrusion Detection System

Sun, Wu et Pooch [40] ont proposé un IDS comportemental qui divise le réseau en zones comme illustré sur la Figure 13, les nœuds dans cette architecture peuvent être classés en deux types :
Les nœuds intrazones et les nœuds interzones (ou nœuds passerelle).

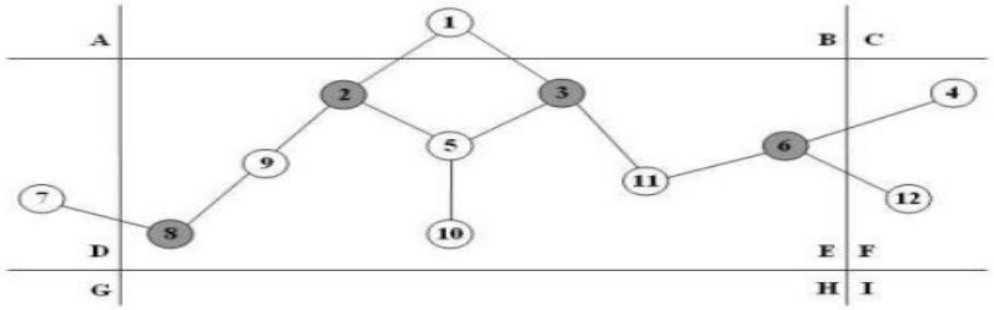


Figure 13: La division du réseau en zone par ZBIDS [40]

En considérant la zone E, les nœuds 5, 9, 10 et 11 sont des nœuds intrazones, tandis que les nœuds 2, 3, 6, et 8 sont des nœuds interzones connectés aux nœuds d'autres zones. La formation et l'entretien des zones exigent que chaque noeud connaisse son emplacement physique pour le mapper sur une carte.

Chaque noeud a un agent IDS semblable à celui proposé par Zhang et Lee. En outre, on y trouve le module *Local Aggregation and Correlation Engines* (LACE) qui combine les résultats des différents modules locaux et qui génère des alertes si un comportement anormal est détecté. Ces alertes sont diffusées à d'autres nœuds dans la même zone.

c.5 Système de détection d'intrusion hiérarchique

La forte mobilité des réseaux VANETs fait qu'une hiérarchie statique n'est pas appropriée pour une telle topologie dynamique de réseau. Stern et al. [41] ont proposé un IDS clustérisé qui est basé sur la signature d'attaque qui peut être structuré en plusieurs niveaux comme le montre la Figure 14.

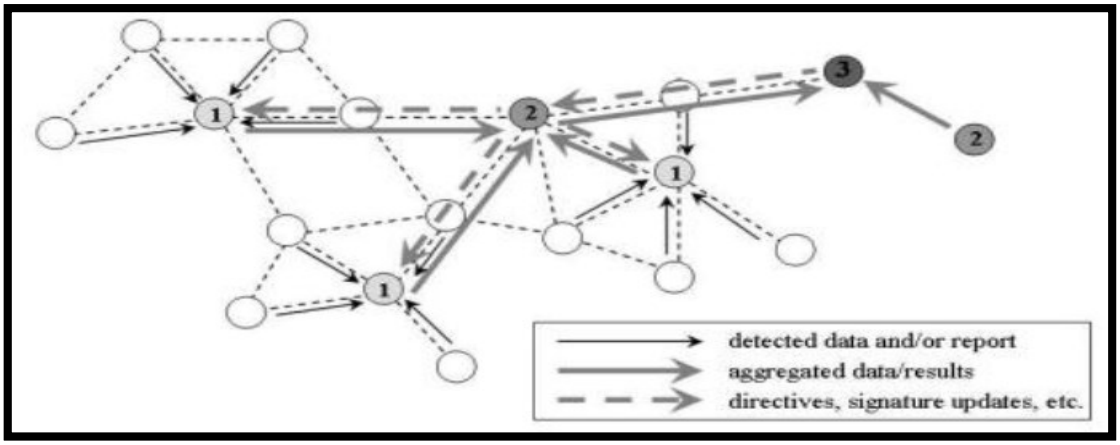


Figure 14: Le système de détection d'intrusion hiérarchique [47]

Les nœuds étiquetés "1" composent le premier niveau qui est constitué de chefs de clusters (clusterheads), les nœuds étiquetés "2" composent le second niveau qui est constitué de chefs de clusters ainsi de suite.
Chaque noeud possède des responsabilités de surveillance (par l'accumulation de statistiques), d'enregistrement, d'analyse (par exemple, contrôler si la signature d'attaque correspond aux têtes des paquets et des payloads), de répondre aux intrusions détectées s'il y a suffisamment de preuves, et d'alerter les chefs de grappe. Les chefs de clusters, en outre, doivent également effectuer :

- Fusion / intégration/ réduction des données
- Détection d'intrusion : les chefs de clusters consolident les données afin de détecter les attaques car un noeud ne pourrait pas être en mesure de détecter une attaque comme DDoS tout seul.
- Gestion de la sécurité : les nœuds les plus élevés de la hiérarchie ont l'autorité et la responsabilité de gestion de détection d'intrusion et d'intervention aux clusters en dessous d'eux. Ils peuvent envoyer des mises à jour de signatures ou des directives et des politiques visant à modifier les configurations pour la détection d'intrusion et la réponse.

c.6 Comparaison des IDS étudiés

Le système de détection d'intrusion	Architecture	La méthodologie	Techniques
Watchdog and Pathrater	Autonome	Surveillance de nœuds relais qui assure le routage	- Auto observation de voisin - Evite les nœuds malveillants dans la recherche de route
CORE	Distribuée et coopérative	Réputation	- la théorie des jeux - Auto observation de voisin - Détection d'égoïsme - Punition de noeud malveillant
IDS de Zhang et Lee	Distribuée et coopérative	Détection coopérative	- Détection locale indépendamment des autres nœuds - Détection globale coopérative si les informations locales sont insuffisantes pour faire des décisions sur une activité.
ZBIDS	Clustérisé	Détection coopérative	- Les agents mobiles - Chaine de Markov - Module LACE et GACE
IDS de Sterne	Clustérisé et hiérarchique	Signature	- Fusion / intégration/ réduction des données -Détection d'intrusion par les chefs de grappe - Gestion de la sécurité

Tableau 1 : Comparaison des systèmes de détection d'intrusion

2.4 Conclusion

Après avoir étudié certaines attaques sur les VANETs, nous avons énuméré les raisons pour lesquelles les solutions de sécurité de routage applicables aux réseaux traditionnels ne le sont pas pour les réseaux ad hoc véhiculaires. Parmi ces raisons, nous avons cité la forte mobilité des nœuds dans les VANETs.

Dans la littérature, on y trouve un ensemble de protocoles de routage pour les réseaux ad hoc véhiculaires sécurisés par des méthodes cryptographiques. Ces solutions utilisent soient des mécanismes légers comme l'authentification de bout en bout par MAC en SRP, soient des mécanismes couteux comme TESLA et la signature à clé publique. Ces techniques montrent des limitations comme les coûts calculatoires importants liés à la génération et la vérification des clés dans TESLA, ainsi que la difficulté liée à la mise en place d'une architecture de gestion de clé. Pour renforcer la sécurité de routage dans les réseaux VANETs, des systèmes de détection d'intrusions ont été ajoutés aux mécanismes cryptographiques.

Chapitre 3 - Menaces et solutions de sécurisation pour les protocoles de routage géographique VANETs

3.1 Introduction

L'objectif de ce chapitre est de présenter certains protocoles de routage géographique, plus particulièrement le protocole GPSR et certaines menaces et solutions de sécurité. D'abord, on va présenter quelques protocoles de routage géographique d'une manière brève, puis le protocole GPSR d'une manière détaillée. Ensuite, nous allons citer des attaques pouvant être perpétrées contre ces protocoles et finalement des solutions de sécurité.

3.2 Protocoles de routage géographique VANETs

Les protocoles de routage géographiques sont les plus adaptés pour les réseaux VANETs, puisque le mécanisme de routage se base sur les données géographiques des nœuds. Dans le routage géographique, il est nécessaire que tous les nœuds soient munis d'un moyen de localisation comme GPS [42]. Le routage géographique s'effectue en deux étapes : la première étape consiste à retransmettre le paquet sur un chemin de routage construit à l'intérieur d'une zone déterminée dite zone de retransmission (Forwarding Zone). La deuxième étape consiste à diffuser le paquet aux nœuds à l'intérieur de la région cible représentant la destination (Geocast Region). Le nœud émetteur utilise d'abord un service de localisation tels que GLS (Greedy Location Service) [43], QLS (Quorum-based location service) [44] ou encore Homezone [45] afin de trouver la position géographique du nœud de destination.

Dans cette section, nous procédons à une description de certains protocoles de routage géographique les plus pertinents.

3.2.1 GSR

Le protocole GSR (Geographic Source Routing) [46] choisit un chemin de route vers la destination en utilisant l'algorithme du plus court chemin de Dijkstra [47] avec les informations de la carte GPS. Ce chemin est constitué d'un ensemble de jonctions qui doit être parcouru pour atteindre la destination. Comme aucune information de trafic en temps réel n'est utilisée, le calcul du plus court chemin se fait sans tenir compte de la quantité de véhicules dans chacune des routes. Comme l'illustre la Figure 15, GSR sélectionne des véhicules pour transférer les données à la prochaine intersection et répète ce processus jusqu'à ce que le nœud de destination soit atteint à l'aide de la stratégie *Greedy Forwarding* le long du chemin. Quand il n'y a aucun véhicule disponible dans la route choisie, GSR tente de sélectionner un autre véhicule à l'extérieur de cette route. Ainsi, la stratégie de récupération utilisée est le *Greedy Forwarding*. Dans leurs tests, les auteurs ont utilisé GPS comme service de localisation.

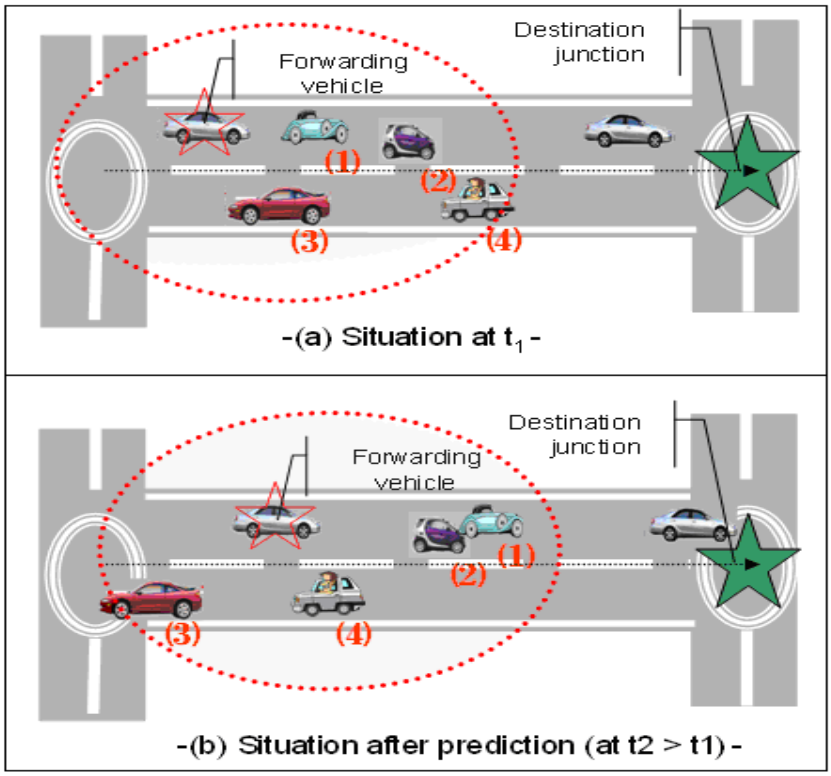


Figure 15: Retransmission des données entre deux jonctions à l'aide de la stratégie greedy améliorée [48]

3.1.1 A-STAR

Le protocole A-STAR (Anchor-based Street Traffic Aware Routing) [49] repose sur le calcul d'un chemin complet pour transmettre les données bien qu'il utilise une approche différente de GSR. Dans A-STAR, le nœud émetteur calcule le chemin de la route (Anchor Path) par l'algorithme de plus court chemin de Dijkstra pondéré par le nombre de lignes de bus qui passent à travers chaque route. Comme pour le cas du protocole GSR, la stratégie de retransmission utilisée le long du chemin est *Greedy Forwarding*. Dans le cas d'un maximum local (Figure 16), le nœud marque la route comme « out-of-service » et recalcule un nouveau chemin de la position courante à la destination, comme le montre la Figure 17 suivante. Quant service de localisation utilisé, les auteurs ne l'ont pas mentionné.

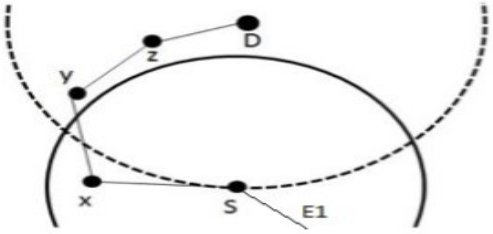


Figure 16: Cas d'un maximum local [50]

Comme nous pouvons le remarquer dans la Figure 16, le nœud source S est plus proche du nœud destinataire D que le nœud voisin (voisin de S) X. Puisque le nœud D ne se trouve pas dans la même zone de retransmission que S, les données provenant de S ne peuvent pas atteindre D en un seul saut. C'est donc une situation de maximum local.

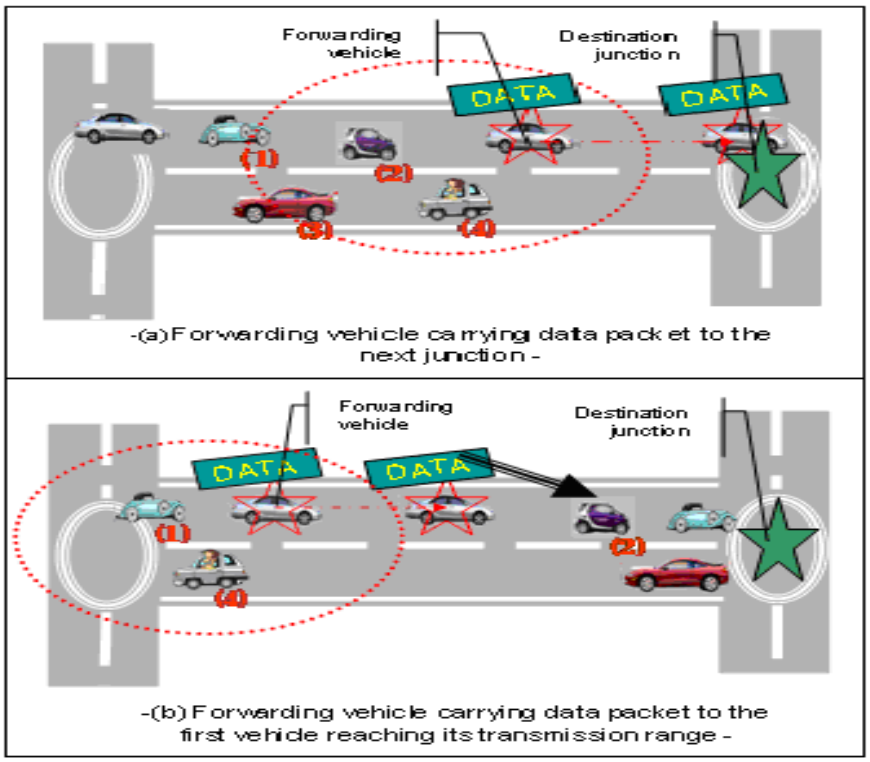


Figure 17: Stratégie de retransmission utilisée dans le cas d'un maximum local [48]

3.1.1 GPSR

GPSR est un protocole de routage réactif et efficace pour les réseaux ad hoc véhiculaires. Contrairement aux protocoles GSR et A-STAR, le protocole GPSR [51] ne calcule aucun chemin de la source à la destination. Dans GPSR, chaque nœud relayeur de paquets insert sa position et celle du nœud destinataire et leurs identifiants (par exemple, leurs adresses IP) dans l'entête d'un paquet de signalisation (messages « beacon ») et l'envoie à ses voisins directs (voisin à un seul saut) en utilisant la stratégie de retransmission *Greedy Forwarding*. Le voisin le plus proche du nœud de destination sera choisi comme le relayeur suivant. L'échange périodique de ces paquets permet aux nœuds de construire leur table de position. La période d'émission des messages « beacon » dépend du taux de mobilité dans le réseau ainsi que de la portée radio des nœuds. En effet, lorsqu'un noeud ne reçoit pas de message « beacon » d'un voisin après un temps T, il considère que le voisin en question n'est plus dans sa zone de couverture et l'efface de sa table de position. Un des avantages des messages « beacon » est qu'un noeud n'a pas besoin que des informations sur ses voisins directs, ce qui nécessite peu de mémoire.

Si aucun voisin n'est plus proche du nœud destinataire par rapport au nœud émetteur, celui-ci passe à un autre mode de retransmission appelé P*erimeter mode,* dans lequel la règle de la main droite est utilisée. GPSR retourne au mode G*reedy Forwarding* si le nœud ayant le paquet admet un voisin plus proche de la destination.
L'acheminement des paquets GPSR selon les deux modes cités précédemment (« Greedy Forwarding » et « Perimeter Forwarding », abrégés respectivement GF et PF) se fait suivant la densité du réseau.

- **Greedy Forwarding**

Le mode GF construit un chemin parcourant les nœuds de la source à la destination où chaque noeud qui reçoit un paquet l'achemine en faisant un saut vers le noeud intermédiaire le plus proche de la destination dans sa zone de couverture. La Figure 18 montre un exemple de ce mode d'acheminement.

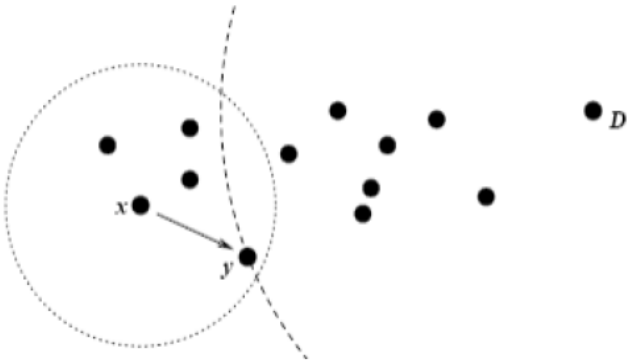


Figure 18: Le voisin de x le plus proche de la destination D est y [51]

- **Perimeter Forwarding**

Le mode « Perimeter Forwarding » consiste à transformer la topologie du réseau en un graphe planaire (ne contenant pas des arrêtes qui se croisent). Le type de graphe peut être RNG (Relative Neighborhood Graph) ou GG (Gabriel Graph) (figure 19).

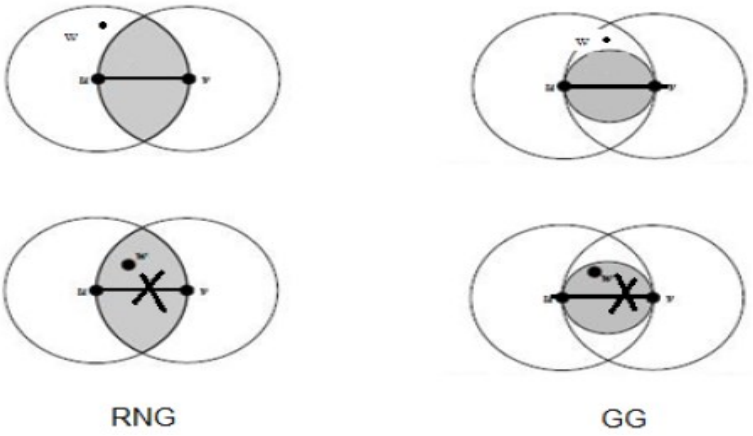


Figure 19: Principe des graphes GG et RNG [52]

Dans un graphe GG ou RNG, deux points sont reliés par une arrête s'il n'y a pas d'autres points dans une certaine « zone interdite » définie mathématiquement par :

- **Relative Neighborhood graph**

$\forall w \neq u, v : d(u, v) \leq \max [d(u, w), d(v, w)]$

- **Gabriel graph**

$\forall w \neq u, v : d^2(u, v) < [d^2(u, w) + d^2(v, w)]$

Le paquet traverse le graphe jusqu'à la destination en utilisant la règle de la main droite « Right-Hand Rule ».

Un paquet GPSR contient dans son entête un champ pour le mode de routage. Ce champ contient « Greedy » lorsque le routage est « Greedy Forwarding » et « Perimeter » lorsque le routage est « Perimeter Forwarding ». Un noeud x recevant un paquet en mode « Greedy » examine sa table de voisins. S'il trouve le voisin le plus proche de la destination alors il lui transmet le paquet. Dans le cas contraire, le noeud va modifier le champ mode de l'entête du paquet par « Perimeter » et enregistre sa localisation. Ensuite, il construit un graphe planaire à partir de ses voisins et transmet son paquet à travers ce graphe. La Figure 19 montre un exemple du chemin emprunté par un paquet GPSR du nœud source x au nœud destination D.

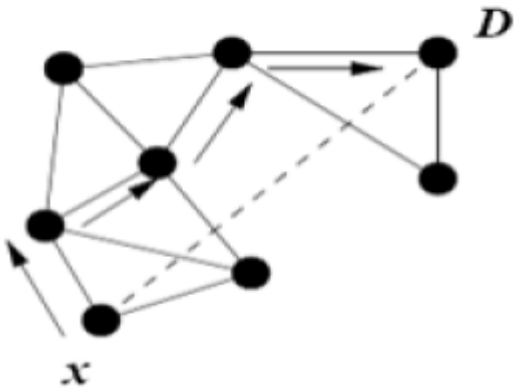


Figure 20: Modes Greedy et Perimeter Forwarding [52]

Sur la Figure 20, le paquet entre en mode Perimeter au nœud x et D est la destination.

Le Tableau 2 présente un résumé et une comparaison des protocoles de routage géographiques précédemment étudiés.

Groupe	GSR	A-STAR	GPSR
Scenario	Sans reconnaissance Avec reconnaissance	Sans reconnaissance Avec reconnaissance	Sans reconnaissance Sans reconnaissance
Architecture	Aucune Aucune	Aucune Aucune	Aucune Aucune
Chemin	Chemin complet Distance	Chemin complet Distance et lignes de bus	Pas de chemin N/a
Retransmission	« Greedy » le long du chemin Non utilisée Sans reconnaissance	« Greedy » le long du chemin Non utilisée Sans reconnaissance	« Greedy » Non utilisée Sans reconnaissance
Récupération	Greedy	Recalcule le chemin	Règle de la main droite

Tableau 2: Comparaison des protocoles de routage géographique étudiés

3.2 Menaces pour les protocoles de routage géographique

Dans cette section, nous allons présenter trois types d'attaques qui concernent spécifiquement les protocoles de routage géographique. Nous commencerons par décrire une attaque qui combine l'attaque de positions et l'attaque Sybil. Ensuite, nous parlerons d'un autre type attaque qui combine une tricherie de positions et une diffusion de fausses positions. Finalement, nous décrirons l'attaque Blackhole surtout sur le protocole GPSR.

3.2.1 Attaques de positions et Sybil

L'attaque de positions peut se produire lorsque les radars présentent une défaillance. Un attaquant peut lancer une attaque de position en modifiant les paquets indiquant la position, en renvoyant des paquets de fausse position et en supprimant des paquets de positions urgentes.

L'attaque Sybil est une attaque nuisible aux VANETs selon laquelle un véhicule prétend être plusieurs véhicules soit en même temps ou successivement. En outre, une attaque Sybil se réfère à une attaque où l'identité du véhicule apparait comme des identités multiples simultanément. L'attaque Sybil est nocive pour les topologies du réseau, les connexions, la consommation de la bande passante du réseau, et elle a même certaines menaces liées à la vie humaine. Un exemple d'une attaque Sybil est montré sur la Figure 21.

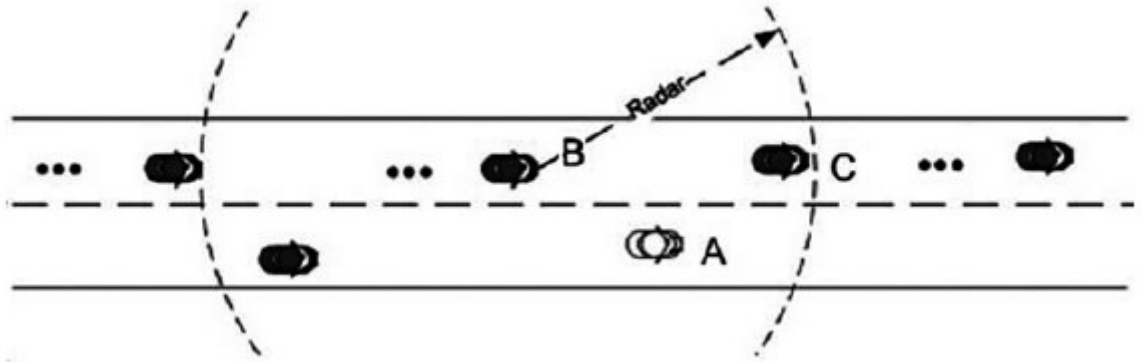


Figure 21: Exemple d'attaque Sybil [29]

Sur la Figure 21, l'attaque de position qui engendre l'attaque Sybil se déroule comme suit : le véhicule A obtient la position P_c du véhicule C. A annonce au victime B que sa position est P_c, et que son identifiant est ID_a. B détecte un véhicule qui est à la position P_c puis conclut que c'est la position de A.

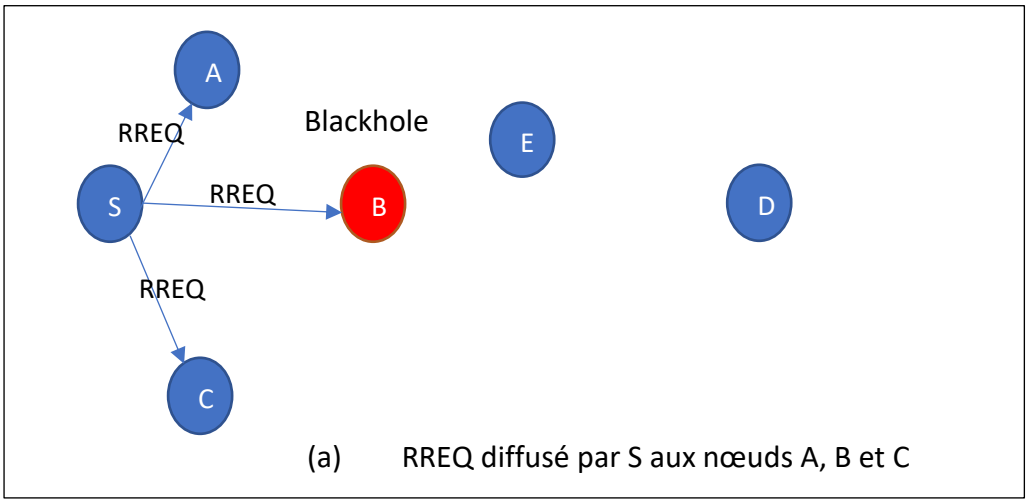
3.2.2 Attaques par tricherie et diffusion de fausses positions

Les réseaux VANETs ont des exigences particulières en termes de mobilité de nœud et des applications dépendant de la position. Ces exigences sont adéquatement couvertes au moyen des protocoles de routage géographique. Les approches du routage géographique sont principalement basées sur les mêmes principes : chaque nœud détermine sa position actuelle au moyen d'un système de positionnement comme un GPS. La position est diffusée périodiquement dans des paquets de balise afin que tous les nœuds au sein de la même zone de transmission sans fil soient en mesure de construire une table de voisins incluant leurs positions. Si un nœud doit transférer un paquet, il sélectionne le saut suivant à partir de la table de voisins, selon une règle prédéfinie (par exemple, il sélectionne le nœud le plus proche de la destination).

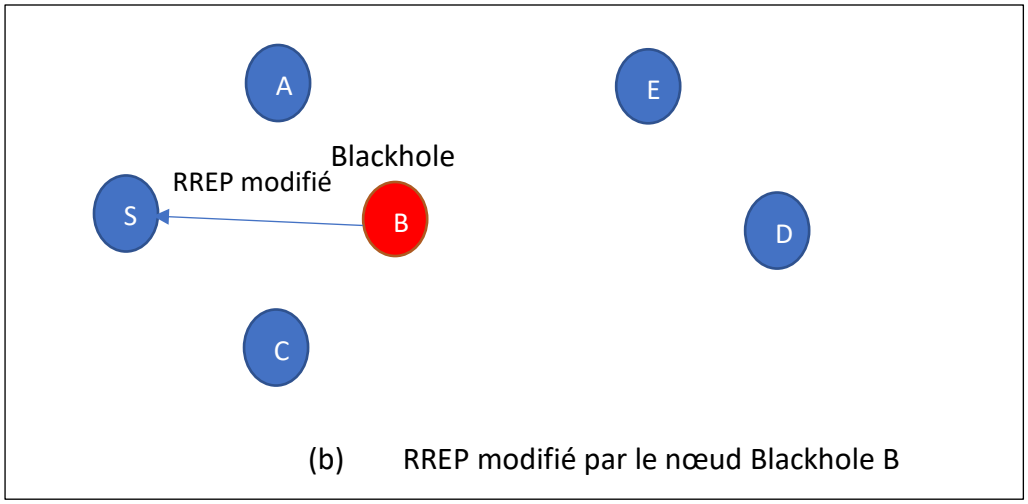
Lorsqu'un nœud diffuse des données de fausse position les messages liés au processus de routage seront affectés. Une Information de fausse position peut être causée soit par un dysfonctionnement au niveau du logiciel de localisation ou une falsification intentionnelle par les attaquants pour rediriger les données. Les nœuds malicieux peuvent dégrader les performances d'un système dans une certaine mesure lorsque le routage des informations à travers de nœuds malicieux viole les objectifs de base de la sécurité tels que la confidentialité, l'authentification, l'intégrité ou la non-répudiation.

3.2.3 Attaque Blackhole sur le protocole GPSR

Dans ce type d'attaque, un nœud malveillant forge le numéro de séquence ou le nombre de saut d'un message de routage (abrégé TTL, pour Time To Live, signifiant le temps de vie d'un paquet) afin d'acquérir une route, ensuite il intercepte et supprime toutes les données qui passent par lui. La Figure 22 représente le comportement d'une attaque Blackhole (attaque du trou noir), dans laquelle le nœud S veut établir une route vers la destination D. Dans le protocole de routage GPSR, en utilisant le mode « Greedy », le noeud S diffusera une RREQ incluant l'ID et la position de la destination pour chercher le noeud le plus proche du nœud de destination D. Ces voisins (ici les nœuds A, B « Blackhole » et C) reçoivent le RREQ, comme le montre la Figure 22 (a) et ils vont lui répondre à travers un RREP, sachant que le noeud Blackhole B à une distance très courte de la destination, comme le montre la Figure 22 (b). Selon la conception de GPSR, le noeud source S va choisir le nœud B (le nœud le plus proche de la destination) pour envoyer les paquets de données. Puisque le nœud B se déclare comme nœud proche de la destination, il est donc considéré comme le saut suivant d'acheminement des paquets vers le nœud destination D. Et lorsque D reçoit les paquets provenant de S, soit il les supprimera directement, soit il modifiera le champ TTL des paquets et les affectent la valeur 0 (TTL=0) et il retransmettra les paquets modifiés vers le saut suivant qui est le nœud E (proche de la destination). Et lorsque le nœud E reçoit des paquets avec un TTL=0, il les supprimera directement.



(a) RREQ diffusé par S aux nœuds A, B et C



(b) RREP modifié par le nœud Blackhole B

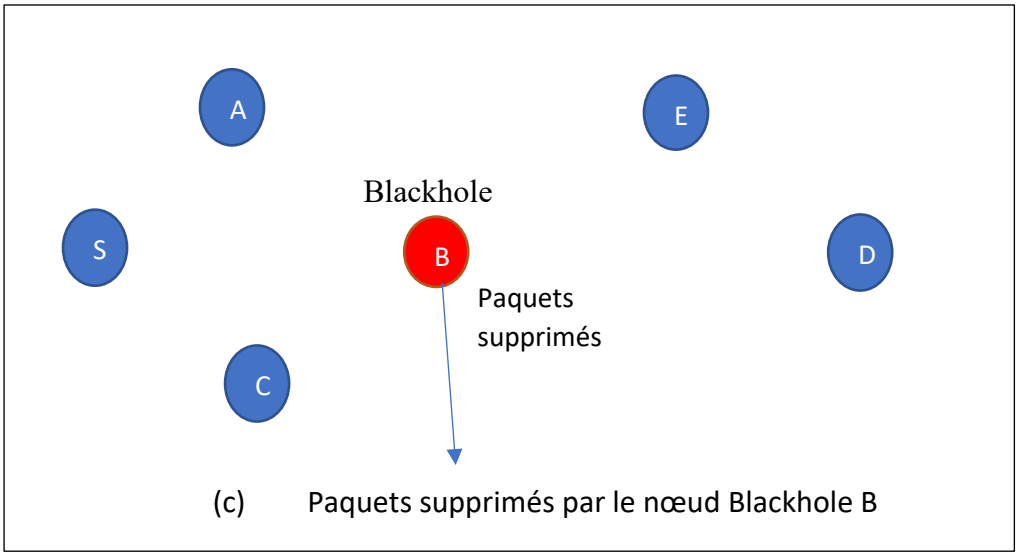


Figure 22: Attaque Blackhole sur le protocole GPSR

3.3 Sécurisation des protocoles de routage géographique

Dans cette partie, nous présentons des solutions de sécurité proposées par des chercheurs pour lutter contre les attaques de positions, Sybil, tricherie et diffusion de fausses positions, ainsi qu'une sécurisation du protocole GPSR.

3.3.1 Sécurisation contre les attaques de positions et Sybil

Pour éviter la plupart des attaques liées à la position et l'attaque Sybil, *Yan* et *al.* [53] ont proposé une nouvelle solution qui a été motivée par la nécessité de fournir des informations de topologie sécurisée dans les VANETs et de construire un réseau sécurisé pour les applications, tel qu'un système d'alerte de congestion. Les auteurs utilisent le radar embarqué comme l'œil virtuel d'un véhicule. Bien que la vue soit limitée à cause d'une portée de transmission radar modeste, un véhicule peut voir autour des véhicules et entendre des rapports de leurs coordonnées GPS. En comparant ce qui est vu avec ce qui a été entendu, un véhicule peut confirmer la position réelle des voisins et isoler les véhicules malveillants pour assurer la sécurité locale. Pour éviter certaines variantes d'attaques Sybil, *Yan et al.* [53] ont proposé une solution selon laquelle si un radar fonctionne d'une façon à ce qu'il peut détecter l'existence physique d'un véhicule, cette information physique peut servir à améliorer les informations hautement abstraites sur le véhicule. Les auteurs ont calculé la similarité entre trois types de données : les détections du radar, les rapports sur le trafic de la circulation en sens inverse et les rapports venant des voisins. Pour mesurer ces similitudes, à chaque similitude est attribuée un poids. Lorsque le radar est en marche, ses détections sont plus dignes de confiance, et par conséquent elles ont un poids plus important. Lorsque le radar n'est pas en marche, ce sont les rapports des voisins qui ont un poids plus important. La position moyenne et la vitesse seront calculées si la similitude est proche.

Un historique de la feuille de route est maintenu en enregistrant ces positions moyennes et les vitesses sur une période de temps. Lorsqu'une requête selon la position doit être lancée, les véhicules restituent l'historique de la carte des véhicules cibles et prennent leurs décisions en se basant sur cette carte.

3.3.2 Sécurisation contre les attaques par tricherie et diffusion de fausse positions

Récemment, *Leinmüller* et *al.* [54] ont adressé la sécurité du routage géographique en proposant une approche basée sur l'espace de conception de base pour la vérification de la position dans les VANETs (représentée sur la Figure 23). Les auteurs ont mis l'accent sur la vérification de positions autonomes sans infrastructure (sous-arbre 1.2.1 Fig. 13). Avec cette approche, chaque nœud juge les revendications de positions indépendamment des autres nœuds. Pour faire ce jugement, l'approche s'appuie entièrement sur des informations de positions qui sont transmises dans des messages beacons réguliers, en supposant que chaque nœud est en mesure de déterminer sa propre position à l'aide d'un système de positionnement (par exemple GPS).

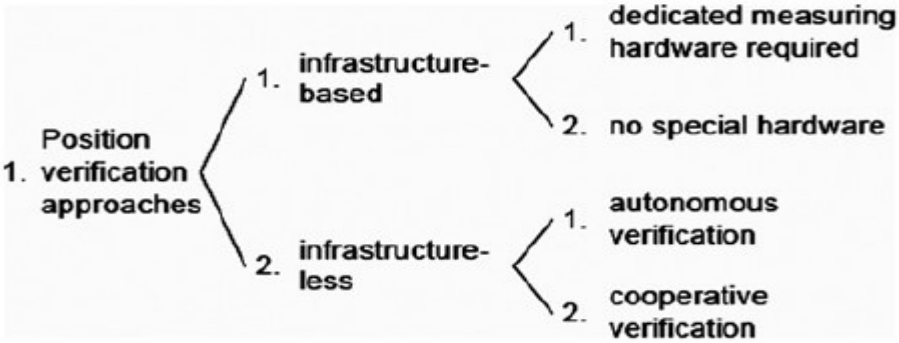


Figure 23: Approches de l'espace de conception pour la vérification de la position

Bien que *Leinmüller* et *al.* ont mis au point des mécanismes visant à détecter et à atténuer l'influence des informations de positions falsifiés dans les protocoles de routage géographiques sans utiliser un matériel spécial (pour mesurer les forces de signal) ou de réseaux d'infrastructures préinstallés, les mécanismes sélectionnés n'empêcheront pas entièrement les nœuds malicieux d'utiliser ces informations de positions falsifiées car le système ne peut pas détecter toutes les fausses positions en raison des faiblesses de la proposition : la solution utilise seulement des seuils durs qui ne sont pas utilisés dans certaines conditions, et le GPS pourrait être coincé ou manipulé. De ce fait, les voitures ne peuvent pas déterminer efficacement leurs propres positions et ne peuvent pas envoyer ou peuvent uniquement envoyer des balises fausses.

Cependant, les mécanismes proposés limiteront considérablement le choix de fausses positions car celles-ci doivent être comprises dans la portée de transmission sans fil d'un nœud. Par conséquent, les possibilités pour les attaquants d'utiliser des fausses positions sont considérablement réduites.

3.3.3 Sécurisation du protocole GPSR

Le protocole GPSR comme tous les autres protocoles de routage géographique a été développé sans tenir compte des aspects de sécurité contre les attaques de routage. Dans cette section, nous présentons une solution de sécurité du protocole GPSR qui est définie en deux étapes :

a. Établissement de clés secrètes

Les chercheurs Erritali, El Ouahidi et Bourget [55] ont proposé une approche permettant d'établir des clés secrètes de Deffie-Hellman entre deux véhicules voisins (en un seul saut) lors de l'échange des paquets balises (en anglais, « beacons »). Ceci permettra de construire des tables de voisins directs pour le routage avec le protocole GPSR [51]. L'objectif est d'avoir des tables de voisins contenant des clés secrètes qui seront utilisées comme des clés de chiffrement symétrique.

b. Ajout d'une signature numérique symétrique

Dans les réseaux Ad hoc véhiculaires, la signature numérique est un mécanisme permettant d'assurer l'intégrité et l'authentification de paquets échangés entre deux voisins GPSR directs. La mobilité des nœuds exige un minimum de temps de routage de paquets de la source à la destination. C'est ainsi que les chercheurs ont proposé d'utiliser l'algorithme de chiffrement symétrique AES [56] au lieu d'un algorithme asymétrique.

La Figure 24 suivante illustre le processus de création de la signature numérique.

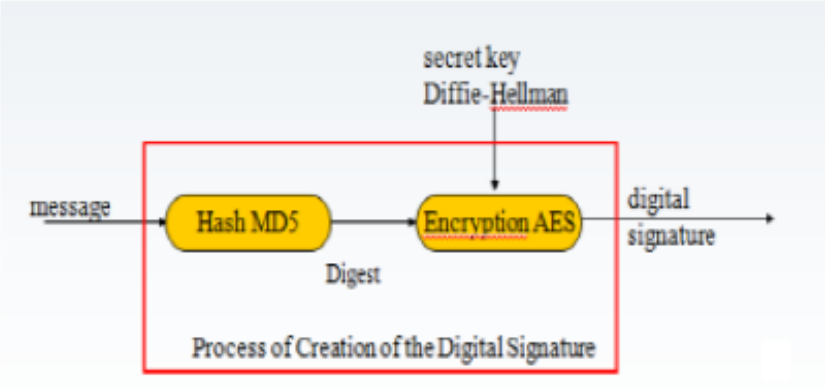


Figure 24: Génération d'une signature numérique [55]

3.4 Conclusion

Dans ce chapitre, nous avons commencé par étudier trois protocoles de routage géographique. Les deux premiers (GSR et A-STAR) se basent sur l'algorithme de calcul du plus court chemin de Dijkstra pour trouver le plus court chemin de la source à la destination avant de commencer l'échange de paquets de données. Par contre, pour le troisième protocole (GPSR), aucun chemin n'est calculé à l'avance. GPSR utilise la méthode « Greedy Forwarding » ou « Perimeter Forwarding » (cas d'un maximum local) pour choisir le nœud voisin de l'émetteur le plus proche de la destination comme relayeur des paquets. Nous avons aussi décrit des menaces et des solutions de sécurité pour les protocoles géographiques, en général et particulièrement l'attaque Blackhole sur le protocole GPSR, ainsi qu'une solution de sécurité.

Chapitre 4 - Simulations et analyse des résultats

4.1 Introduction

Il est évident que parmi les avantages majeurs des réseaux ad hoc véhiculaires c'est qu'ils permettent de sauver des vies humaines en diminuant considérablement le nombre d'accidents. Toutefois, ces mêmes réseaux peuvent être la cause d'une perte de vies humaines dans le cas d'un mal fonctionnement d'un protocole de routage par exemple. En ajoutant à cela le coût financier très élevé pour la mise en place des réseaux VANETs, il est donc important d'avoir une garantie sur le bon fonctionnement du réseau avant son déploiement. C'est pour cette raison que des applications ont été développées pour permettre la simulation des réseaux VANETs.

Dans ce chapitre, nous présenterons d'abord les outils utilisés pour la simulation. Ensuite, nous décrirons les différentes simulations effectuées. Et enfin, nous présenterons les résultats obtenus et leur analyse.

4.2 Outils utilisés pour la simulation

Dans cette section, nous présenterons d'abord les simulateurs de réseaux VANETs, le générateur de mobilité VanetMobiSim, le générateur de scénarios pour NS-2 (NSG2) et l'utilitaire AWK permettant l'extraction des données des paramètres de performances.

4.2.1 Les simulateurs de réseaux

Le nombre de simulateurs de réseaux sans fil ne cesse d'augmenter, ce qui rend plus difficile pour les chercheurs de choisir le plus approprié, répondant le mieux à leurs besoins. Parmi les simulateurs réseaux les plus courants sont NS-2 (Network Simulator 2) [57], Qualnet [58], OPNET [59], OMNeT ++ [60] et GloMoSim [61]. Parmi ces simulateurs, OPNET et Qualnet ne sont pas libres et sont principalement destinés à des fins commerciales. NS-2 a été l'un des simulateurs le plus couramment utilisé durant la dernière décennie. Nous avons utilisé le simulateur NS-2 pour évaluer les performances des protocoles AODV et GPSR sans attaque et dans le cas d'une attaque blackhole. Nous allons donc présenter NS-2 et faire une comparaison par rapport aux autres simulateurs.

a. Le simulateur NS-2

NS-2 est un simulateur de réseau qui bénéficie d'une large communauté d'utilisateurs et de contributeurs, ce qui apporte une grande quantité de documentations et de tutoriels. Il est développé en langage C++ en faisant intervenir des scripts de commande de simulation TCL/OTCL (scripts d'interprétations, évitant la manipulation du C++ aux utilisateurs). Le code source du simulateur est en C++, mais les paramètres de simulations sont définis dans des scripts TCL/OTCL. La simulation est alors lancée en ligne de commande dans un terminal.

Cela permet d'automatiser les simulations de façon simple grâce à des scripts bash (suite d'instructions permettant de faire exécuter plusieurs commandes à la machine). Une interface graphique de visualisation, pouvant présenter les résultats, est fournie avec la version 2.35 : Network AniMator (NAM). Les résultats sont fournis de façon brute dans des fichiers textes qui peuvent être configurés en détail dans les fichiers Tcl. Le traitement des fichiers de résultats se fait à l'aide de scripts externes, tels que des scripts awk (langage de traitement de lignes). Les résultats peuvent alors être exportés dans tout type de logiciels, tels que xgraph, Excel, etc. Dans notre cas, nous avons utilisé LibreOffice Calc, l'équivalent de Microsoft Excel sur Ubuntu pour représenter les paramètres de performances (PDR, délai, gigue, le coût de routage et l'efficacité) sous forme graphique. Le processus d'installation de NS-2 est présenté à l'*Annexe 1*.

- **Fonctionnement de NS-2**

Le simulateur NS-2 est composé de deux éléments fonctionnels (Figure 25) dont :

- **l'interpréteur** qui crée le modèle de simulation.
- **Le moteur de simulation** qui effectue les calculs applicables.

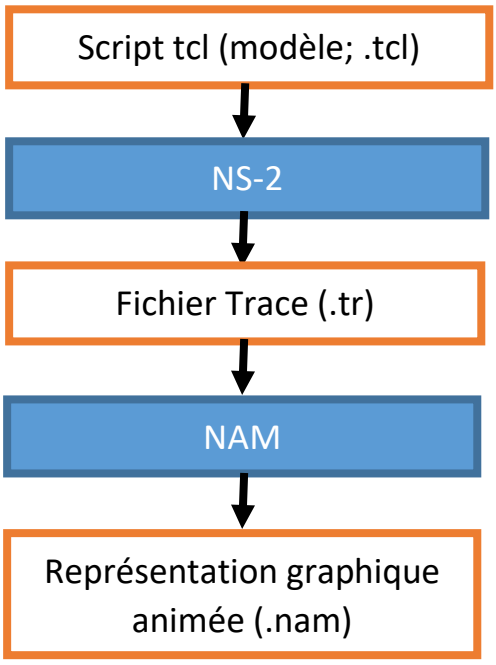


Figure 25: Fonctionnement du simulateur NS-2

- **Protocoles implémentés dans NS-2**

Plusieurs protocoles sont implémentés dans le simulateur NS-2, tels que des protocoles applicatifs, protocoles de transport, protocoles de routage, etc. Dans le cas des protocoles de routage, la majorité des protocoles implémentés sont des protocoles topologiques.
Le tableau 3 représente les différents protocoles implémentés dans NS-2 par rapport aux couches du modèle OSI.

Couche	Protocoles implémentés
Application	Ping, telnet, FTP, multicast FTP, HTTP, vat, générateur de trafic, webcache
Transport	TCP, UDP, SCTP, XCP, TFRC, RAP, RTP Multicast: PCM, SRM, LRM, PLM
Réseau	Unicast: IP, Mobile IP, IPinIP, source routing, nixvector Multicast: SRM, generic centralised Manet: AODV, DSR, DSDV, TORA, IMEP
Liaison	ARP, HDLC, MPLS, GAF, LDP, Diffserv Queueing: dropTail, RIO, RED, WFQ, … MAC: CSMA, CDMA, 802.x, Satellite Aloha, Token ring
Physique	TwoWay, Shadowing, OmniAntennas, EnergyModel, satellite repeater
Support	Générateur de nombre aléatoire, traçage, monitoring, support mathématique, suite de tests, animation (nam), modèles d'erreurs

Tableau 3: Protocoles implémentés dans NS-2 [62]

b. Comparaison des simulateurs de réseaux

Dans le tableau 4, nous comparons les simulateurs de réseaux et leurs caractéristiques.

Simulateurs	GloMoSim	NS-2	NS-3	OMNET++	OPNET	QualNet
Architecture	-	Orienté objet	Orienté objet	Modules	Orienté objet	-
Mobilité	Oui	Oui	Oui	Oui	Oui	Oui
Parallélisme	SMP	Non	Oui	MPI/PVM	Oui	SMP
Modèle d'énergie	-	-	Oui	Oui	Oui	-
Modèle radio	Oui	Oui	Oui	Oui	Oui	-
Interface	Parsec (C)	C++/OTCL	C++/Python	C++	C	Parsec (C)
Licence	Gratuit pour les universitaires	Gratuit	Gratuit	Gratuit pour les universitaires et pour toute utilisation non lucrative	Commercial	Commercial, réductions appliquées pour la recherche
Plateforme	Linux Windows Sun, Mac	Windows UNIX (Linux, Solaris)	Windows UNIX (Linux, Solaris), Mac	Windows (Cygwin) UNIX	Windows (NT, 2000, XP) Solaris	Linux Windows Sun, Mac
Cas d'utilisation	Réseau ad hoc Réseau de capteurs	Réseau ad hoc Réseau filaire	Réseau ad hoc Réseau filaire	N'importe quel type de réseau	Réseaux de capteurs sans fil	Réseaux ad hoc

Popularité (2006)	4%	88%	0% (NS-3 disponible à partir de 2008)	2%	2.6%	2.4%

Tableau 4: Comparaison de simulateurs de réseaux [62]

La modélisation dans NS-2 reste une tâche complexe, car il n'y a pas d'interface graphique et il nécessite souvent une forte technicité pour pouvoir l'utiliser.

Cependant, il a des avantages considérables dont : sa flexibilité, sa richesse, sa réutilisabilité, son extensibilité et la disponibilité de son code.

4.2.2 VanetMobiSim

Le générateur de mobilité VanetMobiSim [63] (Vehicular Ad hoc Network Mobility Simulator) est un ensemble d'extensions de CanuMobiSim, un logiciel permettant de modéliser la mobilité de l'utilisateur utilisé par le groupe de recherche CANU (Communications in Ad Hoc Networks for Ubiquitous Computing) [64], de l'Université de Stuttgart.

Le logiciel CanuMobiSim comprend un certain nombre de modèles de mobilité, ainsi que des analyseurs pour les sources de données géographiques en différents formats et un module de visualisation.

L'ensemble des extensions fournies par VanetMobiSim est composé principalement de deux parties :

- **Un modèle véhiculaire spatial**, composé d'éléments spatiaux (tels que les feux de circulation ou des routes à plusieurs voies), leurs attributs et les relations liant ces éléments spatiaux afin de décrire des zones véhiculaires.
- **Un ensemble de modèles de mobilité axés sur les véhicules**, dont les principales composantes sont le soutien de modèles de mobilité de niveau microscopique :
  - *IDM_IM (Intelligent Driving Model with Intersection Management)*, décrivant parfaitement les gestions de voiture à voiture (car-to-car) et d'intersection.
  - *IDM_LC (Intelligent Driving Model with Lane Changing)*, un modèle de dépassement est également inclus, qui interagit avec IDM_IM pour gérer les changements de voies, les accélérations des véhicules et leurs ralentissements.

VanetMobiSim offre autant de possibilités et de fonctionnalités pour créer des scénarios réalistes. En outre, chaque fonctionnalité contenue dans VanetMobiSim est implémentée en tant que module et est chargée au démarrage à partir d'un fichier de scénario .xml [65], ce qui facilite la configuration de scénario et d'une manière plus pratique. VanetMobiSim est bien plus approprié pour générer des scénarios pour les VANETs que les autres générateurs de mobilité MANET. Les étapes d'installation de VanetMobiSim sont décrites à l'*Annexe 2*.

La Figure 27 présente un système de simulation de VANETs utilisant VanetMobiSim et NS-2.

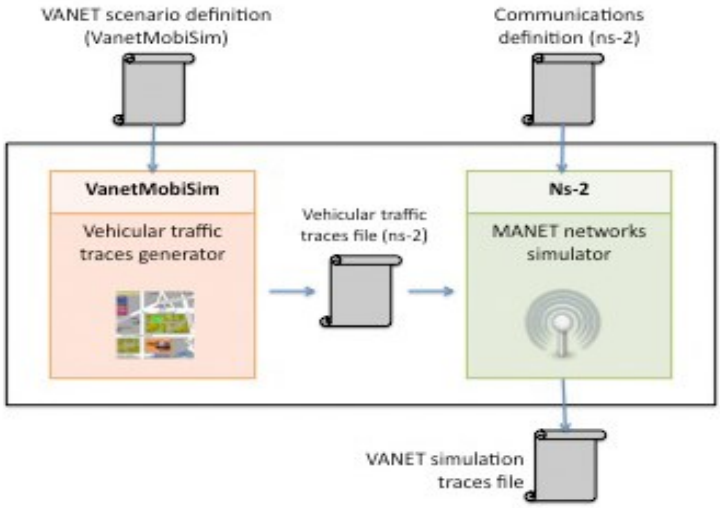


Figure 26: Système de simulation de VANETs utilisant VanetMobiSim et NS-2 [66]

- **Génération de la topologie du réseau**

En supposant que *mobility.xml* est le fichier XML définissant la topologie du réseau, la génération de la mobilité par VanetMobiSim se fait en tapant la commande :

```
#  java -jar VanetMobiSim.jar mobility.xml > fichier_sortie1.tcl
```

Le fichier fichier_sortie.tcl est le fichier de mobilité.

4.2.3 NSG2

Le générateur de scénarios NSG2 pour NS-2 est un outil développé en JAVA par Peng-Jung Wu [67]. Étant donné que NSG2 est écrit en langage JAVA, Il peut être exécuté sur n'importe quelle plateforme. NSG2 est capable de générer des scripts TCL filaires et sans fil pour NS-2. Parmi les fonctions principales de NSG2, on distingue :

1. Création des nœuds pour les réseaux filaires et sans fil.
2. Création de connexions entre les nœuds.
3. Création de liens (Liens Duplexes et liens Simplexes).
4. Création d'agents (TCP et UDP).
5. Création d'applications (CBR et FTP).
6. Mouvement de nœud.

Pour utiliser NSG2, il suffit de télécharger le fichier **jar** via [67] et lancer la commande suivante sur un terminal :

```
#  java  -jar  NSG2.jar
```

NSG2.jar est le fichier jar téléchargé. Il faut donc bien vérifier le nom du fichier et son chemin d'accès.

4.2.4 AWK

L'utilitaire AWK est un outil d'extraction et d'analyse de données qui utilise un langage script basé sur les données. Il a été créé au laboratoire *Bell Labs* en 1970 et son nom dérive des noms de ses auteurs, Alfred **A**ho, Peter **W**einberger et Brian **K**ernighan.
AWK est plus facile à utiliser que la plupart des langages de programmation conventionnels. Il permet de manipuler des chaînes de caractères et d'extraire donc des chaînes particulières comme celles indiquant les valeurs des paramètres de performance (Taux de paquets délivrés, le délai, etc.).

Pour l'installer, il suffit d'exécuter la commande suivante sur un terminal :

```
# sudo apt-get install gawk
```

En supposant que le fichier à traiter se nomme *perf_file.tr* et que le fichier script awk traitant ce fichier et définissant les données à extraire est noté *script.awk*, l'extraction de données se fait en utilisant la commande :

```
# awk -f script.awk perf_file.tr > fichier_sortie2.txt
```

fichier_sortie2.txt est le fichier de sortie dans lequel les données voulues seront enregistrées.

4.3 Description des simulations

Cette section vise à présenter les scénarios et les paramètres utilisés pour implémenter les simulations. À l'Annexe, le code source pour une simulation complète est présenté, mettant l'accent sur comment configurer les scénarios et exécuter les simulateurs.

4.3.1 Scénarios

Au cours du processus de simulation, nous avons implémenté trois scénarios. Dans les deux premiers, nous avons simulé l'attaque blackhole sur le protocole AODV et dans le troisième scénario, cette même attaque est appliquée au protocole GPSR. Ces scénarios sont les suivants :

a. Le scénario Manhattan

Dans l'environnement Manhattan, la disposition des routes est prédéfinie. Manhattan présente plusieurs obstacles (bâtiments, arbres), de feux de circulation et des routes stratégiques sur lesquelles le nombre de véhicules est très élevé par rapports à d'autres. Tout cela conduit souvent à de pertes de connexion entre les entités du réseau.

b. Le scénario urbain

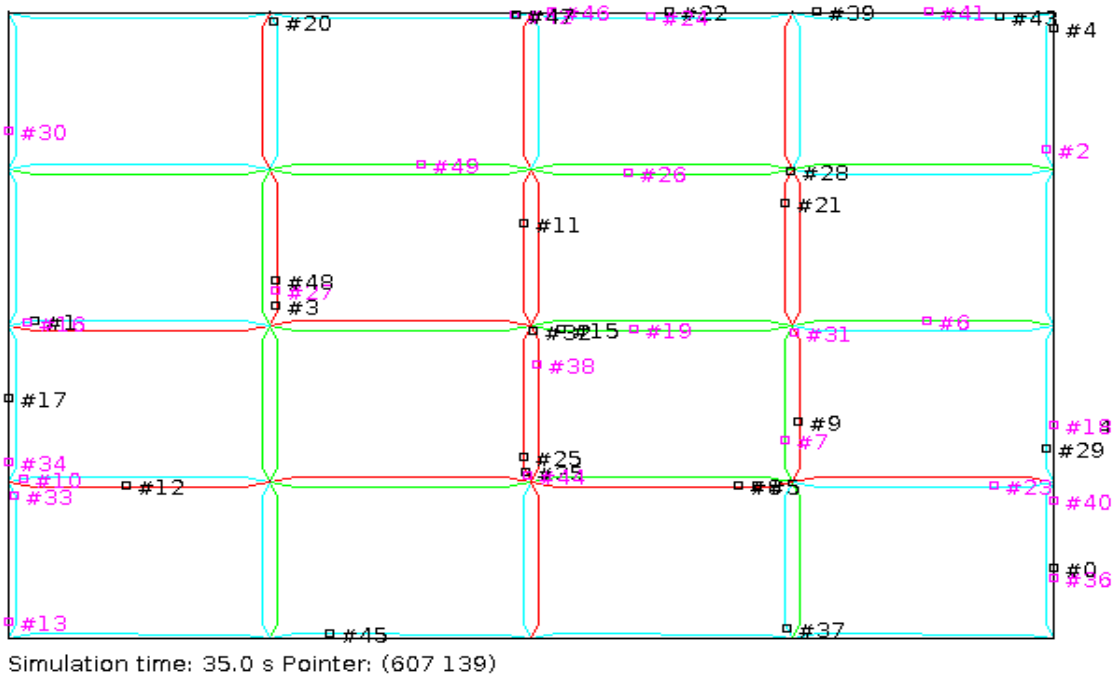
Dans l'environnement de mobilité urbain, la topologie du réseau est plus changeante que dans un environnement Manhattan. Ce changement fréquent permet de rencontrer plus de

véhicules que dans un environnement Manhattan et d'avoir une vue globale même si elle n'est pas très précise. Ceci dynamise l'envoi des messages et augmente leur taux de retransmission.

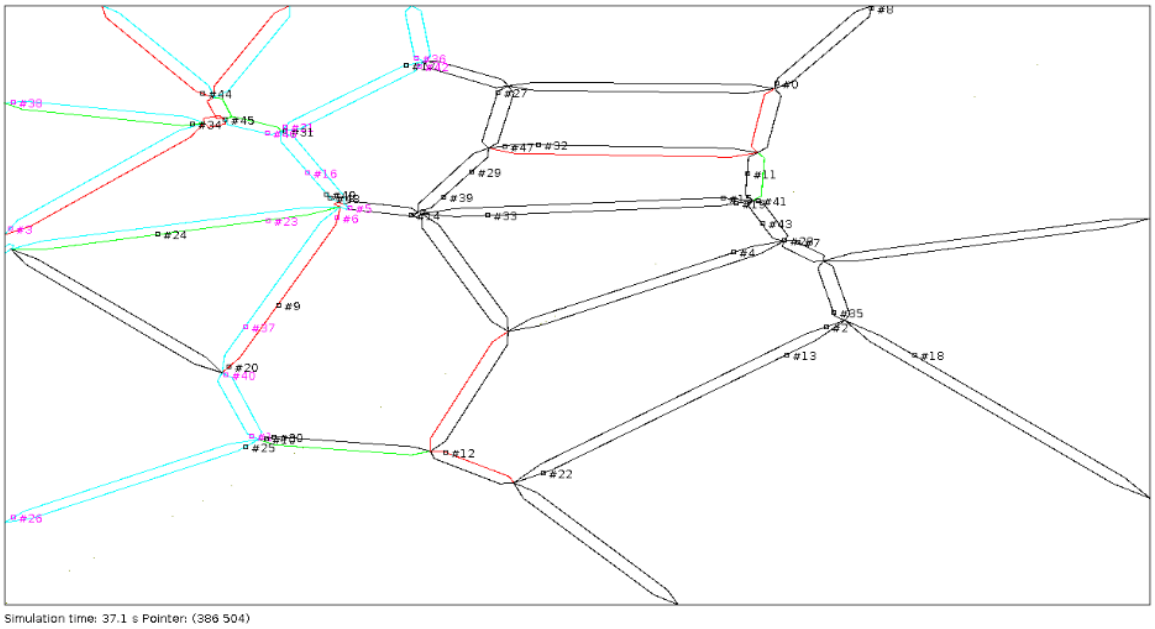
c. Le scénario en Grille

Cette topologie serait en théorie la meilleure. Elle a la forme d'un carré dont deux côtés opposés sont reliés entre eux par des lignes droites. Les nœuds sont positionnés sur les intersections de ces lignes et sur les quatre sommets du carré. Le nombre de nœuds dans les lignes est égale au nombre de nœuds dans les colonnes.

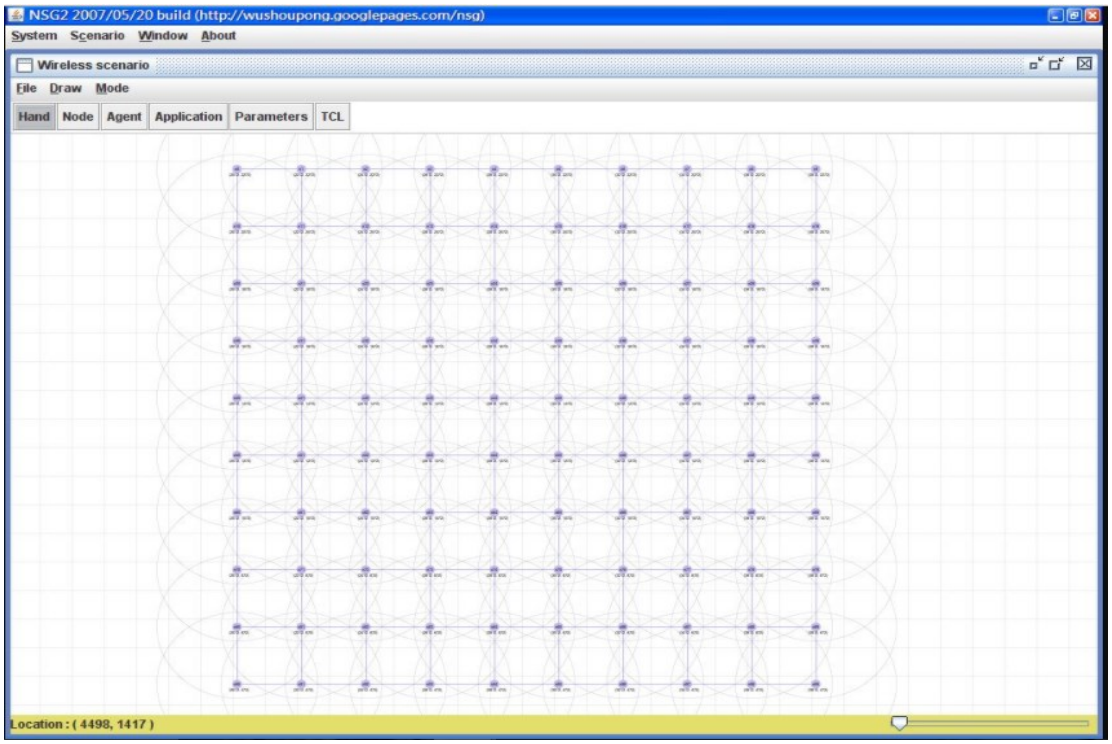
Une image de chaque scénario est présentée dans la Figure 28. Le nombre de nœuds pour les scénarios Manhattan et Urbain est 50.



(a)



(b)



(c)

Figure 27: Scénarios a) Manhattan b) Urbain c) Scénario en Grille

Nous rappelons que nous avons utilisé la topologie en Grille pour simuler l'attaque blackhole sur le protocole GPSR. Nous avons voulu simuler une attaque blackhole sur le protocole GPSR dans les environnements Manhattan et Urbain, mais cela n'a pas été possible.

En effet, en plus de la mobilité des nœuds dans les VANETs, la communication dans le protocole GPSR n'est pas multi-sauts (seuls les nœuds définis dans le fichier de trafic interviennent dans la communication, de la source à la destination). Donc, n'étant pas définis dans le fichier de trafic, les nœuds intermédiaires ne peuvent ni émettre, ni recevoir de paquets.

Ce qui veut dire que dans les environnements Manhattan et Urbain, il est difficile, voire même impossible de trouver un nœud intermédiaire légitime pour simuler une attaque blackhole sur le protocole GPSR.

4.3.2 Paramètres des simulations

Le Tableau 5 présente les valeurs utilisées et la configuration réseau des simulations dans NS-2 pour chaque scénario.

Paramètres	**Valeurs**		
Scénario	Manhattan	Urbain	Scénario en Grille
Générateur de mobilité	VanetMobiSim	VanetMobiSim	NSG
Temps de simulation	300 s	300 s	50 s
Nombre de nœuds	50	50	25
Nombre maximal de connexions	15	15	< 5
Vitesse des nœuds	[20 Km/h ; 65 Km/h]	[40 Km/h ; 100 Km/h]	Pas de mobilité
Dimensions	1001 x 1001 m^2	1001 x 1001 m^2	1934 x 100 m^2
Type de trafic	CBR/UDP	CBR/UDP	CBR/UDP
Longueur des paquets	512 bytes	512 bytes	512 bytes
Taux de transfert de paquets de données	8 paquets/seconde	8 paquets/seconde	8 paquets/seconde
Queue	PriQueue	PriQueue	PriQueue
Antenne	OmniAntenna	OmniAntenna	OmniAntenna
Protocol MAC	IEEE 802.11p	IEEE 802.11p	IEEE 802.11
Modèle de Propagation	TwoRayGround	TwoRayGround	TwoRayGround
Portée de Transmission	250 m	250 m	250 m
Protocole de routage	AODV	AODV	GPSR
Type d'attaque	Blackhole	Blackhole	Blackhole
Nombre de nœuds malicieux	0 à 5	0 à 5	0 à 3

Tableau 5: Vue d'ensemble des caractéristiques de simulation

4.3.3 Paramètres de performance

Le but des expériences réalisées est d'évaluer l'impact de l'attaque blackhole sur les métriques de performance des protocoles de routage topologique AODV et géographique GPSR.
L'étude de l'attaque blackhole sur les environnements Manhattan et Urbain pour le protocole AODV, nous permettra de faire une comparaison des valeurs des paramètres par rapport aux deux environnements.
Nous avons évalué cinq métriques de performance dont :

- **Le taux de paquet délivrés ou PDR (Packet Delivery Ratio)** : c'est le rapport entre le total des paquets de données reçus au niveau de la destination sur le nombre de paquets de données envoyés par les sources, multiplié par 100.
- **Le délai de bout en bout :** c'est le temps écoulé entre l'envoi d'un paquet par un émetteur et sa réception par le destinataire. Les retards pouvant être causés par la retransmission par les nœuds intermédiaires, les délais de traitement, les délais de la file d'attente, et le retard de propagation dans le réseau sont aussi inclus.
- **La gigue :** elle est définie comme étant la variation de la latence (délai de transit ou retard) au fil du temps. Dans les calculs de cette métrique, les paquets perdus sont ignorés.
- **Le coût de routage :** il s'agit du rapport entre le nombre total de paquets de contrôle de routage envoyé et ceux reçus par tous les nœuds. Cette mesure révèle le degré d'efficacité du protocole de routage. En effet, plus le nombre de paquets de contrôle est élevé, moins le protocole est efficace.
- **L'efficacité :** c'est le rapport entre le nombre total de paquets de données délivrés sur la somme de ces paquets et ceux de contrôle de routage.

4.3.4 Implémentation et simulation de l'attaque blackhole

a. Implémentation de l'attaque blackhole

L'implémentation de l'attaque blackhole ne dépend pas de l'environnement de simulation, mais elle diffère d'un protocole à un autre. Dans le cas du protocole AODV, nous nous sommes basés sur le numéro de séquence (numéro permettant de déterminer un chemin à jour vers la destination) pour rendre légitime le nœud blackhole dans le réseau. En effet, un nœud qui présente un numéro de séquence élevé est supposé légitime et meilleur pour retransmettre les paquets vers la destination. Pour que le nœud blackhole soit choisi comme relayeur de paquets vers la destination, nous avons incrémenté le numéro de séquence à sa consultation et le nombre de sauts est mis à 1 pour faire croire à la source que le nœud blackhole est très proche de la destination.

Deux fichiers pour chaque protocole sont à modifier pour simuler l'attaque blackhole. Il s'agit des fichiers *aodv.h* et *aodv.cc* pour le protocole AODV et *gpsr.h* et *gpsr.cc* pour GPSR.

Les instructions d'implémentation de l'attaque sont sélectionnées et sont comprises entre les symboles « //### » ouvrant et fermant.

- **Cas du protocole AODV**

- *aodv.h* : on a déclaré une variable *malicious* de type boolean comme le montre la portion de code suivante :

```
271         /*
272          * History management
273          */
274
275         double          PerHopTime(aodv_rt_entry *rt);
276
277 //### Added for Blackhole Attack - Hafidhou Ibrahim Ahmed Said ###
278         bool            malicious;
279 //#################################################################
280
281         nsaddr_t        index;                  // IP Address of this node
282         u_int32_t       seqno;                  // Sequence Number
283         int             bid;                    // Broadcast ID
284
285         aodv_rtable         rthead;                 // routing table
286         aodv_ncache         nbhead;                 // Neighbor Cache
287         aodv_bcache          bihead;                 // Broadcast ID Cache
288
289         /*
290          * Timers
291          */
292         BroadcastTimer  btimer;
293         HelloTimer      htimer;
294         NeighborTimer   ntimer;
295         RouteCacheTimer rtimer;
296         LocalRepairTimer lrtimer;
```

- *aodv.cc :* dans ce fichier, nous avons d'abord initialisé la variable *malicious* à *false* au niveau du constructeur. Puis, nous avons vérifié que le message passé en paramètre indiquant que le nœud est blackhole est bien la chaîne de caractères « hacker », pour mettre *malicious* à *true* dans la fonction *command*. Ensuite, dans la fonction *rt_resolve* de gestion du routage, on a vérifié si *malicious* vaut *true* (le nœud est blackhole) et on a supprimé le paquet de routage. Et enfin, dans la fonction *recvRequest* le numéro de séquence est incrémenté lorsque le nœud est blackhole (*malicious = true*), puis le paquet de donnée est supprimé.

  - Le constructeur :

```
141 /*
142    Constructor
143 */
144
145 AODV::AODV(nsaddr_t id) : Agent(PT_AODV),
146                           btimer(this), htimer(this), ntimer(this),
147                           rtimer(this), lrtimer(this), rqueue() {
148
149
150   index = id;
151   seqno = 2;
152   bid = 1;
153
154 //### Added for Blackhole Attack - Hafidhou Ibrahim Ahmed Said ###
155   malicious = false;
156 //#################################################################"####
157
158 LIST_INIT(&nbhead);
159   LIST_INIT(&bihead);
160
161   logtarget = 0;
162   ifqueue = 0;
163 }
```

- Pseudo-code de la fonction *command* :

```
76 int
77 AODV::command(int argc, const char*const* argv) {
78   if(argc == 2) {
79   Tcl& tcl = Tcl::instance();
80
81     if(strncasecmp(argv[1], "id", 2) == 0) {
82       tcl.resultf("%d", index);
83       return TCL_OK;
84     }
85
86 //### Added by Hafidhou Ibrahim Ahmed Said for blackhole attack ###
87 if(strcmp(argv[1], "hacker") == 0) {
88 malicious = true;
89 return TCL_OK;
90 }
91 //#####################################################################
```

- Pseudo-code de la fonction *rt_resolve* :

```
443 /*
444   Route Handling Functions
445 */
446
447 void
448 AODV::rt_resolve(Packet *p) {
449 struct hdr_cmn *ch = HDR_CMN(p);
450 struct hdr_ip *ih = HDR_IP(p);
451 aodv_rt_entry *rt;
452
453 //### Added for Blackhole Attack - Hafidhou Ibrahim Ahmed Said ###
454 // If the node is a malicious node, then drop the packet and specify a reason for dropping it!
455 //(Can't openly say you are malicious :-) )
456    if (malicious == true ) {
457       drop(p, DROP_RTR_ROUTE_LOOP); // DROP_RTR_ROUTE_LOOP is added for no reason.
458       return;
459    }
460 //#########################################################
```

- Pseudo-code de la fonction *recvRequest* :

```
777    // Just to be safe, I use the max. Somebody may have
778    // incremented the dst seqno.
779    seqno = max(seqno, rq->rq_dst_seqno)+1;
780    if (seqno%2) seqno++;
781
782
783    sendReply(rq->rq_src,           // IP Destination
784              1,                    // Hop Count
785              index,                // Dest IP Address
786              seqno,                // Dest Sequence Num
787              MY_ROUTE_TIMEOUT,     // Lifetime
788              rq->rq_timestamp);    // timestamp
789
790    Packet::free(p);
791  }
792
793 //### Added for Blackhole Attack - Hafidhou Ibrahim Ahmed Said ###
794  else if (malicious == true) {
795     seqno = max(seqno, rq->rq_dst_seqno)+1;
796     if (seqno%2) seqno++;
797
798     sendReply(rq->rq_src,           // IP Destination
799               1,                    // Hop Count is set to 1 to confuse the source node!
800               rq->rq_dst,           // Dest IP Address
801               seqno,                // Dest Sequence Num
802               MY_ROUTE_TIMEOUT,     // Lifetime
803               rq->rq_timestamp);    // timestamp
804    Packet::free(p);
805  }
806 //#################################################################
```

- **Cas du protocole GPSR**

- *gpsr.h* : la variable *malicious* est déclaré de type entier (*int*) comme le montre le pseudo-code suivant :

```
148 protected:
149   Trace *tracetarget;              //for routing agent special trace
150   void trace(char *fmt,...);       //   Not necessary
151
152   void hellotout();                //called by timer::expire(Event*)
153   void querytout();
154
155 //### Added for Blackhole Attack - Hafidhou Ibrahim Ahmed Said ###
156   int   malicious;
157 //#####################################################################
158
159 public:
160   GPSRAgent();
161
162   int command(int, const char*const*);
163   void recv(Packet*, Handler*);          //inherited virtual function
164
165 };
```

- *gpsr.cc :* dans ce fichier, nous avons d'abord initialisé la variable *malicious* à *0* au niveau du constructeur. Ensuite, nous avons vérifié que le message passé en paramètre indiquant que le nœud est blackhole est bien la chaîne de caractères « hacker », pour mettre *malicious* à *1* dans la fonction *command*. Et enfin, dans la fonction *recv,* le TTL (durée de vie du paquet) est mis à 1, puis il est décrémenté (TTL=0) et le paquet est supprimé.

  - Le constructeur :

```
104 /*
105 * The Constructor
106 */
107 GPSRAgent::GPSRAgent() : Agent(PT_GPSR),
108          hello_timer_(this), query_timer_(this),
109          my_id_(-1), my_x_(0.0), my_y_(0.0),
110          recv_counter_(0), query_counter_(0),
111          query_period_(INFINITE_DELAY)
112 {
113   bind("planar_type_", &planar_type_);
114   bind("hello_period_", &hello_period_);
115
116 ////////Added for Blackhole Attack - Hafidhou Ibrahim Ahmed Said//////
117   malicious = 0;
118 ////////////////////////////////////////////////////////////////////////
119
120   sink_list_ = new Sinks();
121   nblist_ = new GPSRNeighbors();
122
123   for(int i=0; i<5; i++)
124     randSend_.reset_next_substream();
125 }
```

- Pseudo-code de la fonction *command* :

```
388 int
389 GPSRAgent::command(int argc, const char*const* argv){
390   if(argc==2){
391     if(strcasecmp(argv[1], "getloc")==0){
392       getLoc();
393       return TCL_OK;
394     }
395
396 //### Added for Blackhole Attack - Hafidhou Ibrahim Ahmed Said ###
397   if(strcmp(argv[1], "hacker") == 0) {
398   malicious = 1;
399   return TCL_OK;
400   }
401 //#####################################################################
402
403     if(strcasecmp(argv[1], "turnon")==0){
404       turnon();
405       return TCL_OK;
406     }
407
408     if(strcasecmp(argv[1], "turnoff")==0){
409       turnoff();
410       return TCL_OK;
411     }
412
```

- Pseudo-code de la fonction *recv* :

```
340
341   if(cmh->ptype() == PT_GPSR){
342     struct hdr_gpsr *gh = HDR_GPSR(p);
343     switch(gh->type_){
344     case GPSRTYPE_HELLO:
345       recvHello(p);
346       break;
347     case GPSRTYPE_QUERY:
348       recvQuery(p);
349       break;
350     default:
351       printf("Error with gf packet type.\n");
352       exit(1);
353     }
354   }
355
356 //### Added by Hafidhou Ibrahim Ahmed Said for blackhole attack ###
357     /* packet I'm forwarding.
358        Check the TTL. If it is zero, then discard. */
359
360     else {
361     if(malicious==1)
362     iph->ttl_ = 1;
363     if (--iph->ttl_ == 0) {
364       drop(p, DROP_RTR_TTL);
365       return;
366     }
367   /// or drop here. It's the same ///
368    /* iph->ttl_--;
369     if(iph->ttl_ == 0){
370       drop(p, DROP_RTR_TTL);
371       return;
372     }*/
373    forwardData(p);
374   }
375 }//End of the recv function
```

Dans la fonction *recv*, les instructions de configuration de l'attaque sont comprises entre les écrits « //### Added by …. */ » et « /// or drop here …. /// ».

Remarque : Il faut taper la commande *make* après avoir terminé la configuration.

b. Simulation de l'attaque blackhole

Nous avons effectué trois simulations dont la première sur l'environnement Manhattan et la deuxième sur l'environnement Urbain, toutes les deux pour le protocole AODV. La troisième simulation est réalisée sur l'environnement en Grille pour le protocole GPSR. Les paramètres de performances (Taux de paquets délivrés, délai, gigue, coût de routage et efficacité) sont évalués en faisant varier le nombre de nœuds blackholes de 0 à 5 pour les deux premières simulations et de 0 à 3 pour la troisième. Les caractéristiques des simulations sont présentées dans le tableau 5.

Remarque : Le long de la simulation, tout fichier ou sous-répertoire dont le chemin d'accès n'est pas spécifié est supposé être dans le répertoire *ns-allinone-2.35/* où NS-2 est installé. C'est le répertoire courant.

- **Cas du protocole AODV**

Nous rappelons que dans les deux scénarios Manhattan et Urbain, le nombre de nœuds est fixé à 50, le nombre maximal de connexions est 15 et le nombre de nœuds blackholes varie de 0 à 5.

Scénario Manhattan

Pour simuler l'attaque blackhole en utilisant l'environnement Manhattan, on a suivi les étapes suivantes :

1. Dans le fichier *Manhattan.xml* définissant la topologie de cet environnement, le champ *n* de la balise *nodegroup* est affecté par la valeur 50 (le nombre de nœuds). Le code complet du fichier Manhattan.xml est à l'*Annexe 3*.

2. Sur un terminal, nous avons généré le fichier de mobilité *scen-50.tcl* à l'aide du générateur de mobilité VanetMobiSim (l'exécutable : *VanetMobiSim.jar*) en tapant la commande suivante :

```
#  java -jar VanetMobiSim.jar Manhattan.xml > ./scen/scen-50.tcl
```

3. Nous avons généré le fichier de trafic *cbr-50-15-8.tcl* de la commande qui suit :

```
#  ns ./ns-2.35/indep-utils/cmu-scen-gen/cbrgen.tcl -type cbr -nn 50 -seed 1 -mc 15 -rate 8.0 > cbr-50-15-8.tcl
```

4. Lancement de la simulation en utilisant le fichier de simulation *rp-AODV.tcl* (le code complet est dans l'*Annexe 5*) par cette commande :

```
#  ns rp-AODV.tcl AODV 50 15 8 20 20 300
```

Dans les deux dernières commandes précédentes, 50 correspond au nombre de nœuds, 15 indique le nombre maximal de connexions, 8.0 ou 8 représente le taux de paquets transmis par seconde (le débit). Dans la dernière commande, AODV indique le protocole utilisé, la première valeur 20 indique le temps de pause, la deuxième indique la vitesse maximale et 300 correspond au temps de simulation. Si la simulation passe avec succès, les deux fichiers *AODV-50-15-8.tr* permettant d'extraire les métriques de performance et *AODV-50-15-8.nam* permettant de visualiser l'animateur du réseau seront générés.

5. Extraction des métriques de performance en se servant du fichier *perf.awk* (code complet à l'*Annexe 6*) par la commande :

```
# awk -f perf.awk AODV-50-15-8.tr
```

6. Visualisation de l'animateur à l'aide de la commande :

```
# nam AODV-50-15-8.nam
```

- *Scénario Urbain*

On a suivi les mêmes étapes de simulation que le scénario Manhattan en changement quelques paramètres :

1. Remplacer le fichier *Manhattan.xml* par *Urbain.xml* (le code complet est à l'*Annexe 4*) et garder la valeur du champ *n* de la balise *nodegroup* à 50.

2. Taper la même commande que celle de l'étape 2 du scénario Manhattan en remplaçant le fichier *Urbain.xml* par *Manhattan.xml* et *scen-50.tcl* par *scen-50-urbain.tcl*.

3. Si le fichier cbr-50-15-8.tcl est déjà généré, on ne fait rien. Sinon on tape la même commande que celle de l'étape 3 du scénario Manhattan.

4. Même commande que celle de l'étape 4 du scénario Manhattan.

5. Même commande que celle de l'étape 5 du scénario Manhattan.

6. Même commande que celle de l'étape 6 du scénario Manhattan.

- **Cas du protocole GPSR**

Pour le protocole GPSR, le scénario en Grille est utilisé à la place de Manhattan et Urbain. Le nombre de nœuds est fixé à 25, le nombre de nœuds blackholes varie de 0 à 3 et nous avons 5 nœuds sources et un seul nœud destination.

- *Scénario en Grille*

Pour simuler l'attaque blackhole sur le protocole GPSR en utilisant l'environnement en Grille, nous avons le script de simulation *rp-GPSR.tcl* présenté à l'*Annexe 4*. Il suffit d'exécuter la commande suivante sur un terminal :

```
# ns rp-GPSR.tcl
```

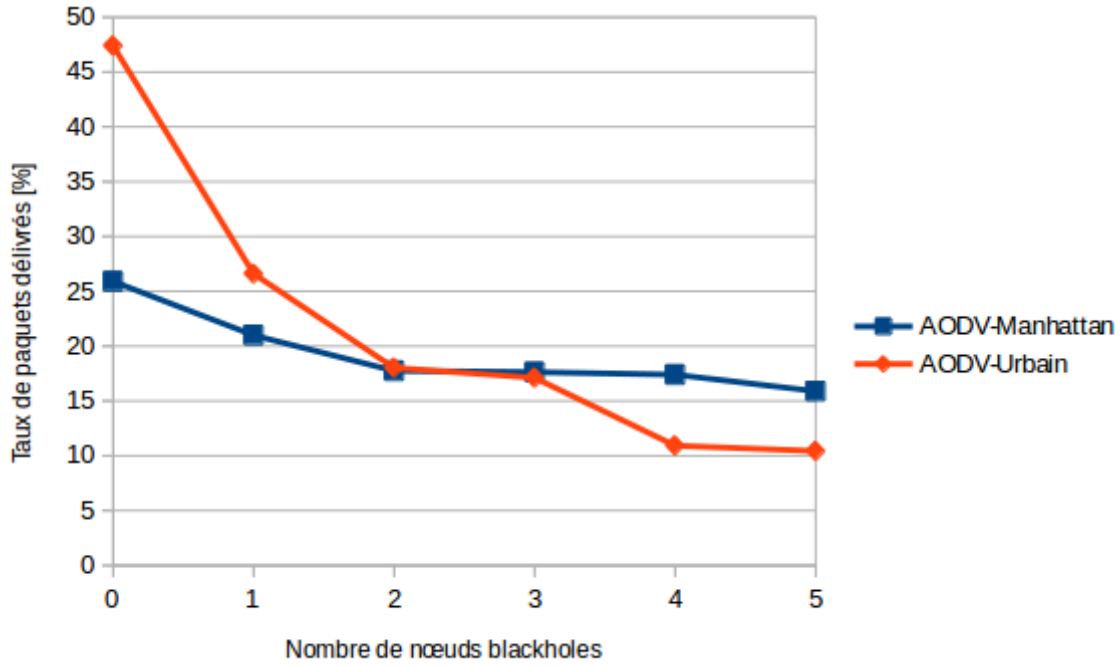
Remarques :

- La topologie en Grille est générée en utilisant l'outil NSG2.1, sachant que le nombre de nœuds est 25.
- La définition de la topologie du réseau et du trafic sont inclus dans le fichier script de simulation *rp-GPSR.tcl*.
- L'extraction des paramètres de performance et la visualisation de l'animateur, se font de la même façon que pour les scénarios Manhattan et Urbain.

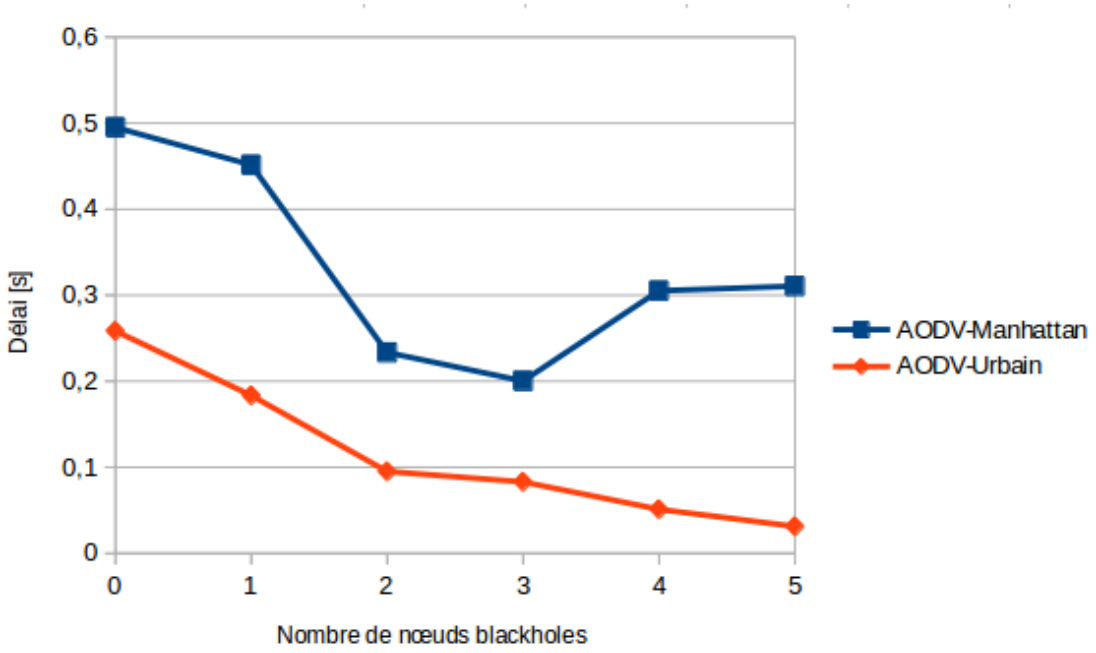
4.3.5 Résultats et analyse

a. Protocole AODV

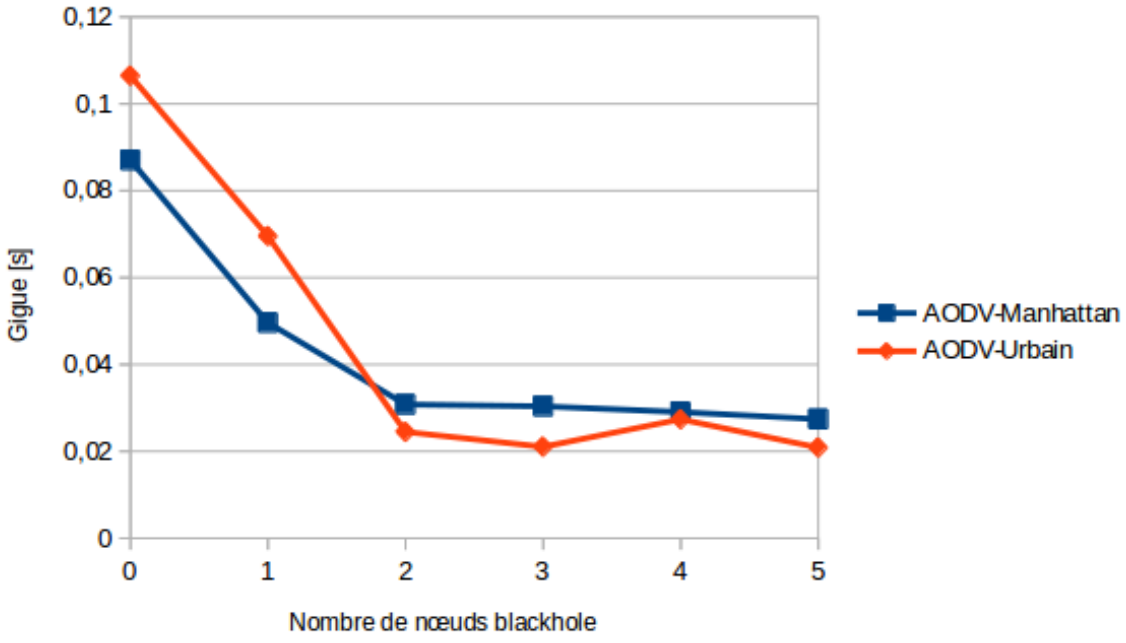
- **Résultats des simulations**



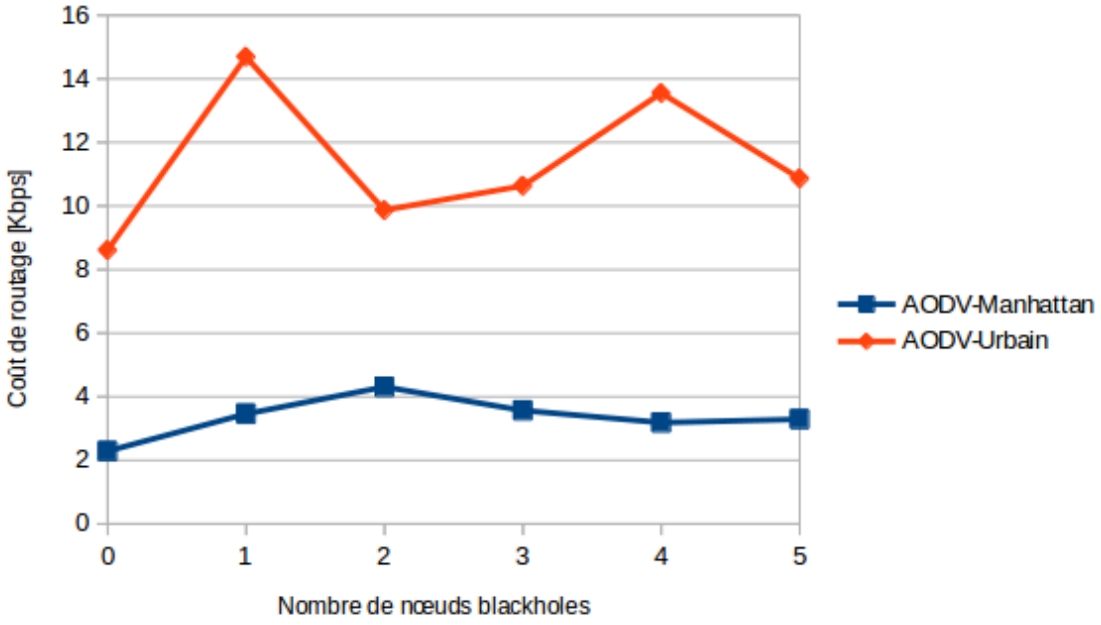
(a) Taux de paquets délivrés par rapport au nombre de nœuds blackholes



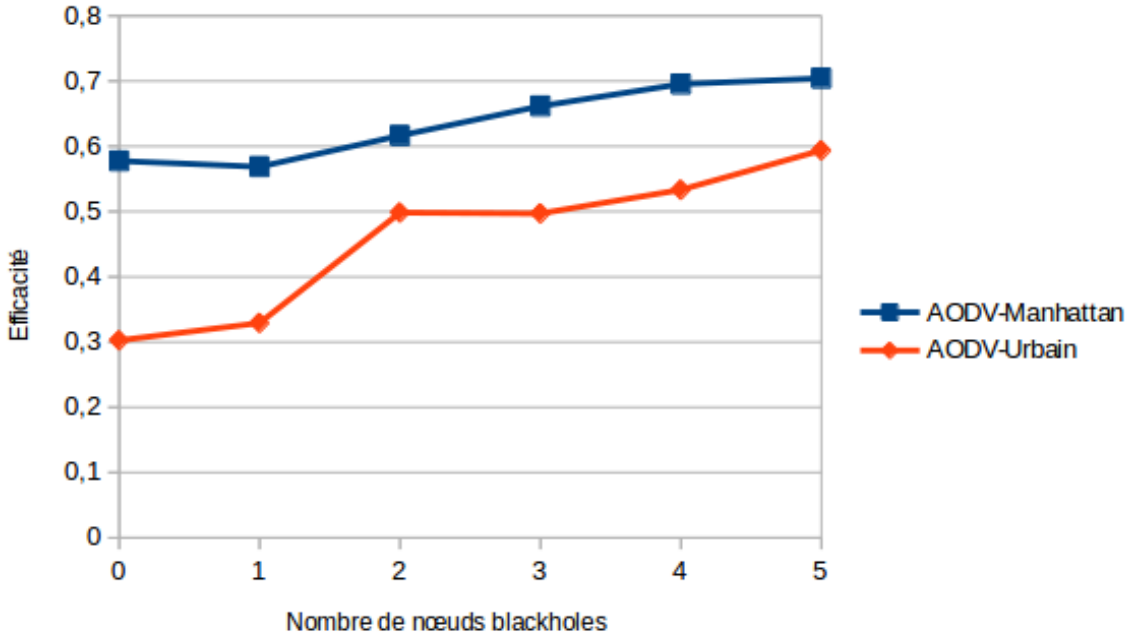
(b) Le délai de bout en bout par rapport au nombre de nœuds blackholes



(c) La gigue par rapport au nombre de nœuds blackholes



(d) Le coût de routage par rapport au nombre de nœuds blackholes



(e) L'efficacité par rapport au nombre de nœuds blackholes

Figure 28: L'impact de l'attaque blackhole sur le protocole AODV dans les scénarios Manhattan et Urbain

Interprétation et analyse des résultats

La Figure 29 - (a) présente une évaluation comparative du taux de paquets délivrés avec succès (PDR) par rapport au nombre de nœuds blackholes pour le protocole AODV dans les scénarios Manhattan et urbain. En analysant cette figure, on remarque que lorsqu'il n'y a aucun nœud blackhole (le nombre de nœud blackhole vaut 0), le PDR dans le scénario Urbain très élevé (environ 48 %) par rapport au scénario Manhattan, qui vaut à peu près 26 %. Lorsque le nombre de nœuds blackholes passe à 1, le PDR dans le scénario diminue brusquement en passant de 48 % à 27 %. Par contre, dans le scénario Manhattan, il ne diminue que de 5 % (de 26 % à 21 %). Lorsque le nombre de nœuds blackhole est 2, le PDR vaut la même valeur 18 % dans les deux scénarios. Ce qui veut dire qu'il a diminué de 30 % dans le cas du scénario Urbain et de 8 % dans le cas du scénario Manhattan. En augmentant d'avantage le nombre de nœuds blackhole, on aperçoit que dans le scénario Manhattan le PDR reste presque constant, tandis qu'il continue à diminuer dans le scénario Urbain et en plus la courbe passe au-dessous de celle du scénario Manhattan à partir du nombre de nœuds blackhole égale à 3.

Ces résultats s'expliquent par le fait que la densité des nœuds et leur forte mobilité dans le scénario Urbain font que des nœuds intermédiaires sont très rapidement consultés lorsqu'il n'y a pas d'attaque (PDR élevé) et lors des attaques, les nœuds malicieux sont beaucoup plus fréquentés (PDR très bas) que dans le scénario Manhattan.

Le délai de bout en bout est illustré par la Figure 29 – (b). Dans le scénario Urbain, plus le nombre de nœuds blackholes augmente, plus le délai diminue. Par contre, dans le scénario Manhattan, le délai diminue lorsque le nombre de nœuds blackhole est entre 0 et 3 et augmente quand le nombre de nœuds passe de 3 à 5. En effet, par la forte mobilité des nœuds dans le scénario Urbain et par le fait que les déplacements ne sont pas limités (cas contraire du scénario Manhattan), les nœuds blackholes sont beaucoup plus convoités. Et plus un nœud blackhole reçoit de paquets, plus le délai diminue car il y aura moins de paquets qui arriveront à destination.

La gigue (en anglais, jitter) est représentée sur la Figure 29 – (c). Comme ce fut le cas du PDR, on peut remarquer à travers cette figure que lorsque le nombre de nœuds attaquants est nul, la courbe de la gigue dans le scénario Urbain est au-dessus de celle qui correspond au scénario Manhattan. Mais lorsqu'on augmente le nombre de nœuds blackholes et surtout à partir de deux nœuds blackholes, la courbe de la gigue dans le scénario Urbain passe au-dessous de celle du scénario Manhattan. Cela montre bien que plus le nombre de nœuds blackholes est élevé, plus la variation de délais de transit et des retards des paquets arrivant à la destination est beaucoup plus faible dans le scénario Urbain que dans le scénario Manhattan. Ceci est causé par la forte consultation des nœuds blackholes dans le scénario Urbain que dans le scénario Manhattan.

Remarque :

Les résultats obtenus pour le délai et la gigue semblent être un peu absurdes. Cependant, pour ces deux paramètres (le délai et la gigue), seuls la variation de latence et le temps mis par les paquets arrivés à destination (ceux des paquets supprimés sont ignorés dans les calculs) sont comptabilisés. C'est pour cela que les valeurs de ces deux paramètres diminuent au lieu d'augmenter.

Le coût de routage et l'efficacité sont illustrés respectivement par les Figures 29 – (d) et 29 – (e). Ces deux métriques de performances sont toujours complémentaires. En effet, dans la Figure 29 – (d), la courbe du coût de routage dans le scénario Urbain se situe au-dessus de celle correspondant au scénario Manhattan, mais c'est le cas contraire des courbes représentant l'efficacité dans la Figure 29 – (e). Ce résultat justifie toujours la prépondérance des nœuds blackholes dans le scénario Urbain beaucoup plus que dans le scénario Manhattan. Ainsi, nous pouvons dire que plus le nombre de paquets reçu diminue, plus le coût de routage augmente et l'efficacité du protocole diminue.

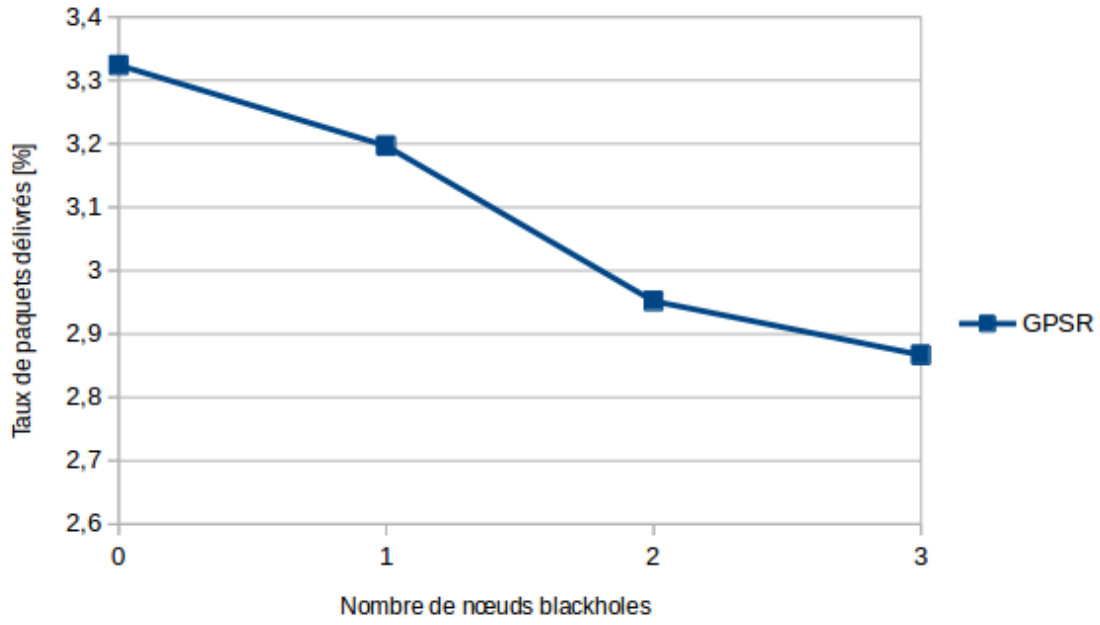
En faisant une analyse globale de ces résultats, nous pouvons conclure que l'environnement Urbain et beaucoup plus sensible aux attaques et particulièrement à l'attaque blackhole qu'à l'environnement Manhattan. Le Tableau 6 présente une interprétation des résultats obtenus. La variable *n* indique le nombre de nœuds blackholes.

Métriques d'évaluation des performances du protocole AODV	Scénario Manhattan	Scénario Urbain
Taux de paquets délivrés	Moyen sans attaque et bas si *n* est élevé	Très élevé sans attaque et très bas si *n* est élevé
Délai de bout en bout	Très élevé sans attaque, très bas à *n={1; 2; 3}* et élevé à partir de *n=4*	Moyen sans attaque et très bas si *n* est élevé
Gigue	Très élevé sans attaque, très bas à *n={1; 2} et* presque constante à *n={3; 4; 5}*	Très élevé sans attaque, très bas à *n={1; 2} et* presque constante à *n={3; 4; 5}*
Coût de routage	Bas à *n=0*, élevé à *n=2*, bas à *n=3* et presque constant à partir de *n=4*	Moyen à *n=0*, très élevé à *n=1, très bas à n=2*, *très élevé à n={3; 4} et faible à n=5*
Efficacité	Elevé à *n=0*, presque constant à *n=1*, élevé de *n=2* à *n=4* et presque constant à *n=5*	Moyen à *n=0*, très élevé à *n={1; 2}*, constant à *n=3* et élevé à *n={4; 5}*

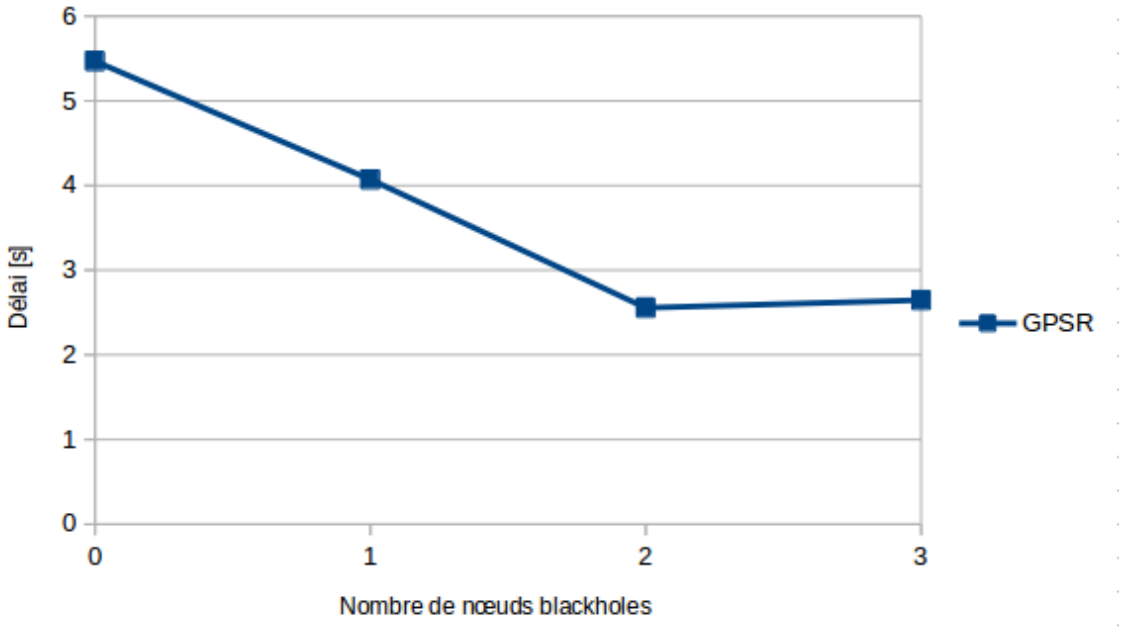
Tableau 6: Interprétation des résultats de la simulation de l'attaque blackhole sur AODV

b. Protocole GPSR

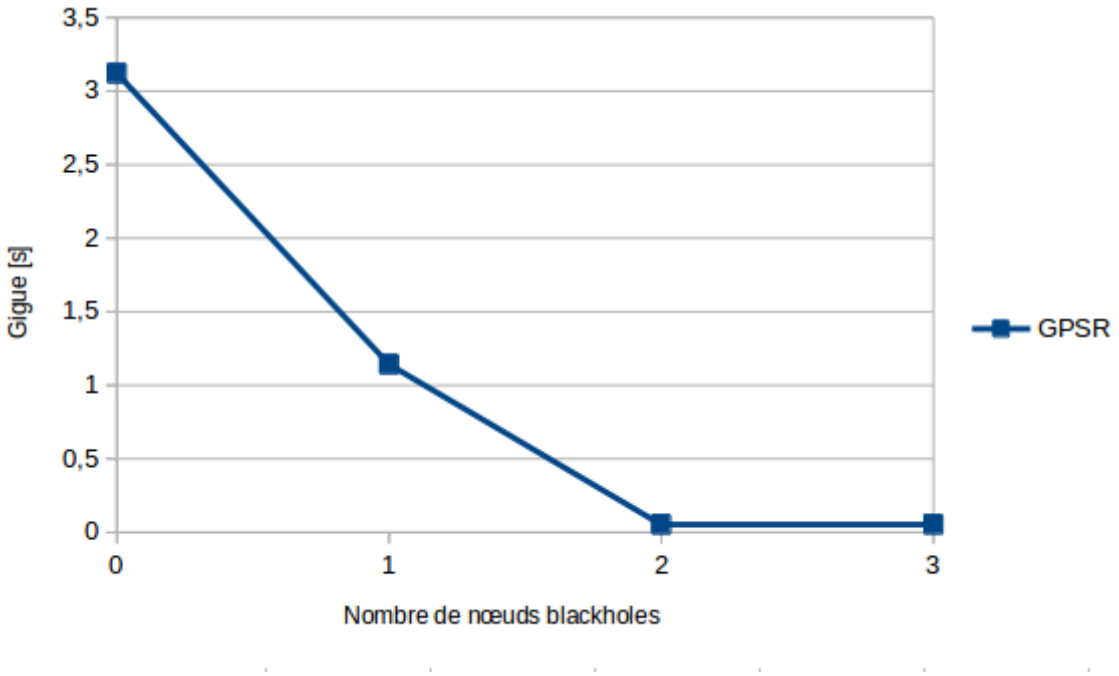
- **Résultats des simulations**



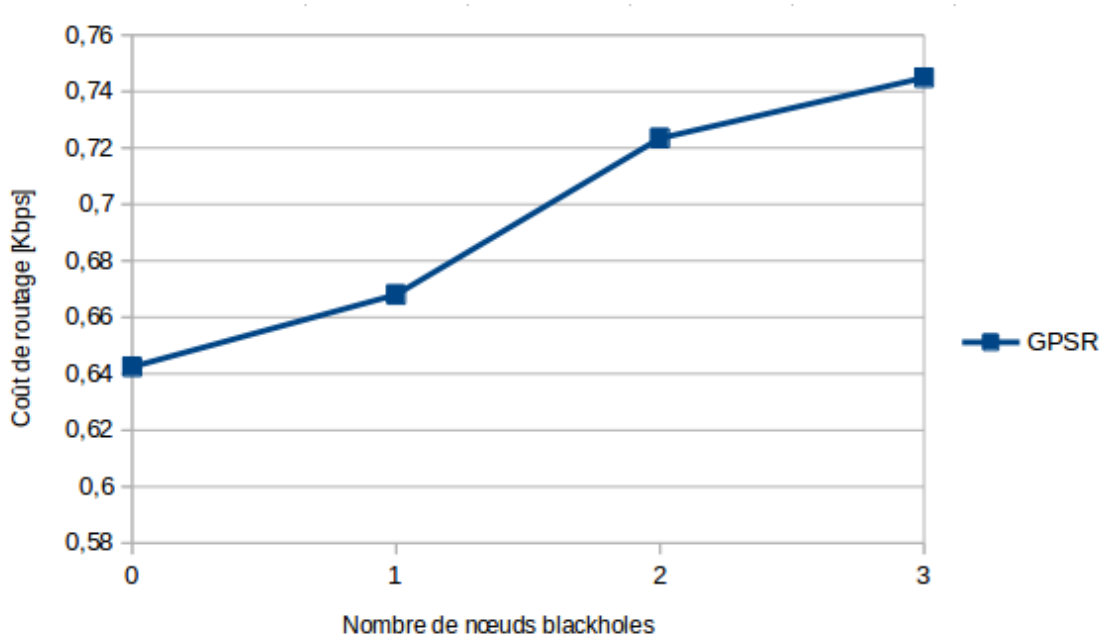
(a) Taux de paquets délivrés par rapport au nombre de nœuds blackholes



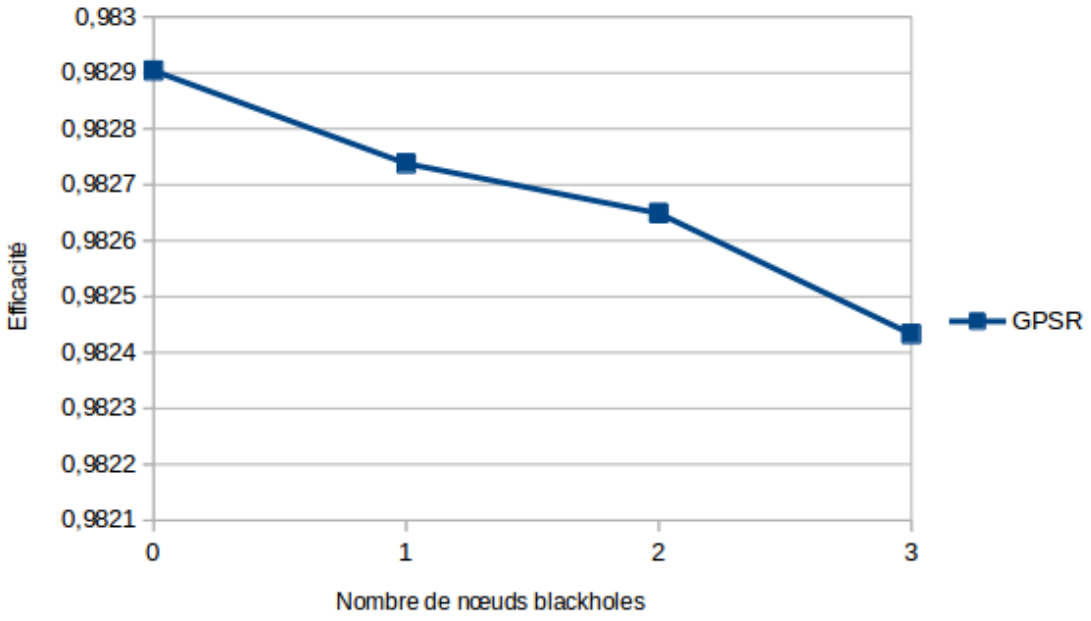
(b) Le délai de bout en bout par rapport au nombre de nœuds blackholes



(c) La gigue par rapport au nombre de nœuds blackholes



(d) Le coût de routage par rapport au nombre de nœuds blackholes



(e) L'efficacité par rapport au nombre de nœuds blackholes

Figure 29: L'impact de l'attaque blackhole sur le protocole GPSR dans le scénario en Grille

Interprétation et analyse des résultats

Nous faisons une analyse presque globale des résultats des simulations de l'attaque blackhole sur le protocole GPSR dans le scénario en grille. En analysant les Figures 30 – (a), 30 – (b), 30 – (c) et 30 – (e) qui représentent respectivement le taux de paquets délivrés, le délai de bout en bout, la gigue et l'efficacité, on remarque que lorsqu'il n'y a pas de nœuds blackholes la valeur de chaque métrique est élevé (on a une bonne performance). Mais plus le nombre de nœuds blackholes augmente, plus la performance de ces paramètres diminue, sauf pour le cas du délai et de la gigue. Pour ces deux paramètres (le délai et la gigue), seuls la variation de latence et le temps mis par les paquets arrivés à destination (ceux des paquets supprimés sont ignorés dans les calculs) sont comptabilisés. C'est pour cela que les valeurs de ces deux paramètres diminuent au lieu d'augmenter. On peut dire qu'un nombre élevé de nœuds blackholes donne lieu à une énorme perte de paquets, ce qui justifie la forte diminution des performances de ces paramètres de qualité de service (QoS).
Quant au coût de routage représenté sur la Figure 30 – (d), plus le nombre de nœuds blackholes augmente, plus il augmente aussi, car il y a moins de paquets reçus.

On comprend très bien de ces résultats que l'attaque blackhole diminue considérablement les performances des paramètres de qualité de service du protocole GPSR dans les réseaux VANETs. Ce qui nécessite des mécanismes de sécurité pour remédier aux vulnérabilités de GPSR.

Nous résumons ces résultats dans le Tableau 7 suivant. La variable *n* indique le nombre de nœuds blackholes.

Métriques d'évaluation des performances du protocole AODV	Scénario en Grille
Taux de paquets délivrés	Très élevé à *n=0* et très bas à *n>0*
Délai de bout en bout	Très élevé *n=0*, très bas à *n={1; 2}* et presque constant à *n=3*
Gigue	Très élevée n=0, très basse à *n={1; 2} et* presque constante (nulle) à *n=3*
Coût de routage	Bas à *n=0*, très élevé à *n>2*
Efficacité	Très élevé à *n=0* et très basse à *n>0*

Tableau 7: Interprétation des résultats de la simulation de l'attaque blackhole sur GPSR

4.4 Conclusion

Dans ce chapitre expérimental, nous avons commencé par étudier les outils dont on s'est servi pour réaliser les simulations. Nous avons décrit le simulateur réseau NS-2 et nous l'avons comparé à d'autres simulateurs. Le générateur de mobilité VanetMobiSim nous a permis de définir la topologie du réseau et la mobilité des nœuds pour le protocole AODV, dans les scénarios Manhattan et Urbain. Nous avons pu évaluer l'impact de l'attaque blackhole sur les performances des paramètres de qualité de service du protocole AODV. Par contre, pour le protocole GPSR, nous avons rencontré des difficultés lors de la simulation de l'attaque blackhole en utilisant les mêmes scénarios. De ce fait, on a fait recours au générateur de scénarios NSG2 pour générer un scénario en Grille nous permettant de contrôler le trafic du réseau et de mesurer l'impact de l'attaque blackhole sur les performances des paramètres de qualité de service pour GPSR.

Conclusion et perspectives

Ces travaux de recherche nous ont permis d'une part, de nous familiariser avec le monde de la recherche informatique et d'autre part, d'apprendre de nouvelles technologies qui sont les réseaux ad hoc véhiculaires ainsi que de nouveaux outils utiles pour simuler ces réseaux. Nous avons appris à travers ces travaux, l'essor que connaissent les réseaux informatiques en générale et plus particulièrement les réseaux VANETs. Nous avons étudié certaines menaces pour les réseaux ad hoc véhiculaires et des solutions de sécurisation.

Nous avons décrit trois protocoles de routage géographique tels que le protocole GSR, le protocole A-STAR et le protocole GPSR. Les deux premiers protocoles calculent d'abord le chemin de routage le plus court de la source à la destination en utilisant l'algorithme de Dijkstra avant de commencer le processus d'échange de paquets. La forte mobilité des nœuds dans les réseaux VANETs rend ces protocoles moins adaptés aux VANETs par rapport au protocole GPSR. Ce dernier agit directement sur les positions des nœuds et choisit le nœud le plus proche de la destination et qui se trouve dans la même zone de couverture que l'émetteur pour retransmettre les paquets vers la destination. GPSR n'a donc pas besoin de connaître à l'avance les positions de tous les nœuds pour effectuer le routage. C'est ainsi qu'il est considéré comme étant l'un des protocoles de routage géographique les mieux adaptés aux VANETs. Et c'est d'ailleurs à cet effet que nous l'avons choisi pour évaluer l'impact de l'attaque blackhole sur les performances des paramètres de qualité de service.

Nous avons simulé l'attaque blackhole sur le protocole de routage topologique AODV dans les scénarios Manhattan et Urbain. Les résultats obtenus montrent que l'environnement Urbain par la forte mobilité des nœuds est beaucoup plus sensible à l'attaque blackhole que l'environnement Manhattan. Contrairement au protocole GPSR, la configuration et la simulation de l'attaque blackhole sur AODV est plus facile car AODV est déjà implémenté dans NS-2 et des documentations sont disponibles pour la simulation des attaques. Nous nous sommes donc basés sur l'implémentation de l'attaque blackhole sur AODV pour l'implémenter aussi sur GPSR. Pour AODV, nous avons incrémenté le numéro de séquence et mis le nombre de sauts à 1 pour attirer les paquets vers le nœud blackhole. Quant au protocole GPSR, nous avons utilisé la valeur du TTL, qui, quand elle passe à 0, le paquet est supprimé.

Les résultats des simulations réalisées montrent que l'attaque blackhole a considérablement des effets néfastes sur les performances des paramètres de qualité de service du protocole GPSR, comme ce fut le cas du protocole AODV. La plupart des solutions de sécurisation des protocoles de routage géographique, le cas de GPSR par exemple, se basent sur des primitives cryptographiques (de chiffrement, de génération de signature numérique, etc.) qui permettent certes d'assurer l'intégrité et l'authentification des paquets. Toutefois, le nœud blackhole peut ne pas supprimer les paquets lui-même et initialiser le TTL à 1 pour que lorsqu'il retransmet le paquet au nœud suivant, celui-ci décrémentera le TTL (TTL = 0) et supprimera donc les paquets.

Comme perspectives, nous proposons d'explorer d'autres protocoles de routage, de tenir compte d'autres types d'attaques et de proposer des solutions robustes, comme le fait de combiner l'authentification des nœuds et le chiffrement des données.

Références

[1] G. Chandrasekaran, “VANETs: The Networking Platform for Future Vechicular Applications,” *Rutgers Univ.*, pp. 45–51, 2007.

[2] A. C. Networks, “Introduction to Mobile Ad hoc Networks (MANETs).”

[3] S. Petersburg, “2013 13th International Conference on ITS Telecommunications,” no. November, 2013.

[4] H. Te Wu, W. S. Li, T. S. Su, and W. S. Hsieh, “A novel RSU-based message authentication scheme for VANET,” *Proc. - 5th Int. Conf. Syst. Networks Commun. ICSNC 2010*, pp. 111–116, 2010.

[5] P. Papadimitratos *et al.*, “Secure Vehicular Communication Systems: Design and Architecture,” 2009.

[6] M. S. Al-kahtani, “003_SP_Survey on security attacks in Vehicular Ad hoc Networks (VANETs),” *2012 6th Int. Conf. Signal Process. Commun. Syst.*, pp. 1–9, 2012.

[7] S. Jaap, M. Bechler, and L. Wolf, “Evaluation of Routing Protocols for Vehicular Ad Hoc Networks in Typical Road Traffic Scenarios,” *Proc. 11th EUNICE Open Eur. Summer Sch. Networked Appl.*, pp. 584–602, 2005.

[8] T. Iii, P. Sabatier, and P. Lorenz, “These Jonathan Petit-13072011,” 2011.

[9] J. B. Kenney, “Dedicated short-range communications (DSRC) standards in the United States,” *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[10] K. Bilstrup, E. Uhlemann, E. G. Ström, and U. Bilstrup, “Evaluation of the IEEE 802.11p MAC method for vehicle-to-cehicle communication,” *IEEE Veh. Technol. Conf.*, pp. 1–5, 2008.

[11] J. Zhu and S. Roy, “MAC for Dedicated Short Range Communications in Intelligent Transport System,” *IEEE Commun. Mag.*, vol. 41, no. 12, pp. 60–67, 2003.

[12] B. H. Kim, K. Y. Choi, J. H. Lee, and D. H. Lee, “Anonymous and traceable communication using tamper-proof device for vehicular Ad hoc Networks,” *2007 Int. Conf. Converg. Inf. Technol. ICCIT 2007*, pp. 681–686, 2007.

[13] F. Kargl, Z. Ma, and E. Schoch, “Security Engineering for VANETs,” *4th Work. Embed. Secur. Cars escar 2006*, no. December, pp. 1–10, 2006.

[14] A.-S. Pathan Khan, *Self-Organizing Networks - MANET, WSN, WMN, VANET*, CRC Press. New York: Auerbach, 2011.

[15] F. Armknecht and A. Festag, “Cross-layer Privacy Enhancement and Non- repudiation in Vehicular Communication,” no. November, 2015.

[16] B. Parno and A. Perrig, “Challenges in securing vehicular networks,” *Work. hot Top. networks*, no. 4, pp. 1–6, 2005.

[17] Y. Qian and N. Moayeri, “DESIGN SECURE AND APPLICATION-ORIENTED VANET,” *Semant. Sch.*, 2008.

[18] C.-I. Fan, R.-H. Hsu, and C.-H. Tseng, “Pairing-based Message Authentication Scheme with Privacy Protection in Vehicular Ad Hoc Networks,” *Proc. Int. Conf. Mob. Technol. Appl. Syst.*, p. 82:1--82:7, 2008.

[19] I. Aad, J.-P. Hubaux, and E. W. Knightly, “Impact of Denial of Service Attacks on Ad Hoc Networks,” *Networking, IEEE/ACM Trans.*, vol. 16, pp. 791–802, 2008.

[20] J. P. Hubaux and N. Ben Salem, “Securing wireless mesh networks,” *Wirel. Commun. IEEE*, vol. 13, pp. 50–55, 2006.

[21] J. Blum and A. Eskandarian, “The threat of intelligent collisions,” *IT Prof.*, vol. 6, pp. 24–29, 2004.

[22] J. M. De Fuentes, A. I. González-Tablas, and A. Ribagorda, “Overview of Security Issues in Vehicular Ad-hoc Networks,” *Handb. Res. Mobil. Comput.*, pp. 894–911, 2010.

[23] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K. Li, “Security and Privacy Issues for Inter-vehicle Communications in VANETs,” *2009 6th IEEE Annu. Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Networks Work.*, pp. 1–3, 2009.

[24] C. Harsch, A. Festag, and P. Papadimitratos, “Secure position-based routing for VANETs,” *IEEE Veh. Technol. Conf.*, pp. 26–30, 2007.

[25] X. LIN and R. LU, *VEHICULAR AD HOC NETWORK SECURITY AND PRIVACY*. .

[26] N. W. Lo and H. C. Tsai, “Illusion attack on VANET applications - A message plausibility problem,” *GLOBECOM - IEEE Glob. Telecommun. Conf.*, 2007.

[27] S. Ma, O. Wolfson, and J. Lin, “A survey on trust management for intelligent transportation system,” *Proc. 4th ACM SIGSPATIAL Int. Work. Comput. Transp. Sci. - CTS ’11*, pp. 18–23, 2011.

[28] M. Ketel, “Applying the Mobile Agent Paradigm to Distributed Intrusion Detection in Wireless Sensor networks knecessaryvetionuseso mechaism Fof d itrusin e pesmenmhanistri Inethispaery w o,” *Syst. Theory, 2008. SSST 2008. 40th Southeast. Symp. on. IEEE*, pp. 1–5, 2008.

[29] J. T. Isaac, S. Zeadally, and J. S. Cámara, “Security attacks and solutions for vehicular ad hoc networks,” *IET Commun.*, vol. 4, no. 7, p. 894, 2010.

[30] S. Čapkun, L. Buttyán, and J. Hubaux, “Self-Organized Public-Key Management for Mobile Ad Hoc Networks,” *IEEE Trans. Mob. Comput.*, vol. 2, no. 5005, pp. 52–64, 2003.

[31] W. Diffie, P. van Oorshot, and M. Wiener, “Authentication and Authenticated Key Exchange,” *Des. Codes Cryptogr.*, vol. 2, pp. 107–125, 1992.

[32] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks,” *Wirel. Networks*, vol. 11, no. 1–2, pp. 21–38, 2005.

[33] P. G. Argyroudis and D. O’Mahony, “Secure routing for mobile ad hoc networks,” *IEEE Commun. Surv. Tutorials*, vol. 7, no. 3, pp. 2–21, 2005.

[34] M. G. Zapata, “Secure ad hoc on-demand distance vector routing,” *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, p. 106, 2002.

[35] F. Anjum, D. Subhadrabandhu, and S. Sarkar, “Signature based intrusion detection for wireless ad-hoc networks: a comparative study of various routing protocols,” *2003 IEEE 58th Veh. Technol. Conf.*, vol. 3, 2003.

[36] a Mishra, K. Nadkarni, and a Patcha, “Intrusion Detection in Wireless Ad Hoc Networks,” *Wirel. Commun. IEEE*, vol. 11, no. 1, pp. 48–60, 2004.

[37] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” *Proc. 6th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom ’00*, no. September

2014, pp. 255–265, 2000.

[38] P. Michiardi and R. Molva, “CORE : A COLLABORATIVE REPUTATION MECHANISM TO ENFORCE NODE COOPERA TION IN MOBILE AD HOC NETWORKS.”

[39] and W. L. Zhang, Yongguang, “Intrusion Detection in Wireless Ad-Hoc Networks,” *Proc. 6th Annu. Int. Conf. Mob. Comput. networking. ACM*, pp. 275–283, 2000.

[40] B. Sun, K. Wu, and U. W. Pooch, “Alert Aggregation in Mobile Ad Hoc Networks,” *WiSE’03*, no. January 2003, pp. 69–78, 2003.

[41] D. Sterne *et al.*, “A General Cooperative Intrusion Detection Architecture for MANETs,” 2005.

[42] N. Drawil, *Improving the VANET Vehicles ’ Localization Accuracy Using GPS Receiver in Multipath Environments [D]*. Waterloo, Ontario, 2007.

[43] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, “A scalable location service for geographic ad hoc routing,” in *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom ’00*, 2000, pp. 120–130.

[44] Z. J. Haas and B. Liang, “Ad hoc mobility management with uniform quorum systems,” *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 228–240, 1999.

[45] I. Stojmenovic and B. Vukojevic, “A routing strategy and quorum based location update scheme for ad hoc wireless networks,” *Comput. Sci. Univ. Ottawa, Tech. Rep. TR-99-09*, pp. 1–16, 1999.

[46] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, “A routing strategy for vehicular ad hoc networks in city environments,” *Intell. Veh. Symp. 2003. Proceedings. IEEE*, vol. 2000, no. 1, pp. 156–161, 2003.

[47] J. Chen, “Dijkstra’s shortest path algorithm,” *J. Formaliz. Math.*, vol. 15, 2003.

[48] M. Jerbi, S. M. Senouci, R. Meraihi, and Y. Ghamri-Doudane, “An Improved Vehicular Ad Hoc Routing Protocol for City Environments,” in *Communications, 2007. ICC ’07. IEEE International Conference on*, 2007, pp. 3972–3979.

[49] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, “A stable routing protocol to support ITS services in VANET networks,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 6 I, pp. 3337–3347, 2007.

[50] A. Fonseca and T. Vaz??o, “Applicability of position-based routing for VANET in highways and urban environment,” *J. Netw. Comput. Appl.*, vol. 36, no. 3, pp. 961–973, 2013.

[51] B. Karp and H. Kung, “GPSR: Greedy Perimeter Stateless Routing for wireless networks,” *ACM MobiCom*, no. MobiCom, pp. 243–254, 2000.

[52] A. C. Casablanca and B. France, “Contribution à la sécurisation des réseaux ad hoc véhiculaires,” pp. 60–61, 2013.

[53] G. Yan, S. Olariu, and M. C. Weigle, “Providing VANET security through active position detection,” *Comput. Commun.*, vol. 31, no. 12, pp. 2883–2897, 2008.

[54] T. Leinm"{u}ller *et al.*, “Improved security in geographic ad hoc routing through autonomous position verification,” vol. 2006, pp. 57–66, 2006.

[55] M. Erritali, B. El Ouahidi, and D. Bourget, “Secured Geographic Routing Protocol for Vehicular Ad Hoc Networks (VANETs),” pp. 311–315, 2013.

[56] S. Heron, “Advanced Encryption Standard (AES),” *Netw. Secur.*, vol. 2009, no. 12, pp. 8–12, 2009.

[57] K. Fall and K. Varadhan, “The network simulator (ns-2),” *URL http//www. isi. edu/nsnam/ns*, 2007.

[58] “Scalable Network Technologies.” [Online]. Available: http://www.qualnet.com/.

[59] “OPNET, Application and Network Performance.” [Online]. Available: http://www.opnet.com/.

[60] “OMNeT++ Network Simulation Framework.” [Online]. Available: http://www.omnetpp.org/.

[61] “GloMoSim, Global Mobile Information System Simulation Librarym.” [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/.

[62] “Les simulateurs réseaux.” [Online]. Available: http://www.usthb.dz/fei/.

[63] J. Härri, F. Filali, C. Bonnet, and M. Fiore, “VanetMobiSim: generating realistic mobility patterns for VANETs,” *Proc. 3rd Int. Work. Veh. ad hoc networks*, pp. 4–5, 2006.

[64] “CANU Research Group (Stuttgart University).” [Online]. Available: http://canu.informatik.uni-stuttgart.de/mobisim/downloads/.

[65] J. Harri, M. Fiore, F. Filali, and C. Bonnet, “Vehicular mobility simulation with VanetMobiSim,” *Simulation*, vol. 87, no. 4, pp. 275–300, 2011.

[66] “VanetMobiSim/Ns-2 Simulator | VANET/ITS Website (NEO).” [Online]. Available: http://neo.lcc.uma.es/staff/jamal/vanet/?q=node/9.

[67] “NS-2 Scenarios Generator 2 (NSG2).” [Online]. Available: https://sites.google.com/site/pengjungwu/nsg.

[68] “Network Simulator and Network AniMator (nsnam).” [Online]. Available: https://sourceforge.net/projects/nsnam/.

[69] “Fix NS-2 Installation Errors.” [Online]. Available: https://www.quora.com/How-do-I-install-NS2-software.

[70] “VanetMobiSim (Politecnico di Torino).” [Online]. Available: http://vanet.eurecom.fr/.

Annexe 1

Installation de NS-2

Dans cette étude, nous avons installé la version NS-2.35 sur un système Ubuntu 14.04. Pour se faire, il faut d'abord télécharger le fichier de code source *ns-allinone-2.35.tar.gz* depuis [68] et l'extraire à l'aide de la commande suivante dans un répertoire de base au choix, exemple : le répertoire personnel */home/hafidhou* :

```
# tar -xzf ns-allinone-2.35.tar.gz -C ~/
```

Le répertoire */ns-allinone-2.35* sera donc crée sur le répertoire de base spécifié après l'extraction. Avant de commencer l'installation, les paquets *build-essential*, *libX11-dev* et *xorg-dev* doivent éventuellement être installés à l'aide des commandes :

```
# sudo apt-get install build-essential

# sudo apt-get install libX11-dev

# sudo apt-get install xorg-dev
```

Pour commencer l'installation, il faut entrer dans le répertoire */ns-allinone-2.35* et lancer l'installation en tapant les commandes suivantes :

```
# cd ~/ns-allinone-2.35

# ./install
```

Si l'installation est effectuée avec succès, un message apparaîtra sur le terminal comme celui de la Figure 25.

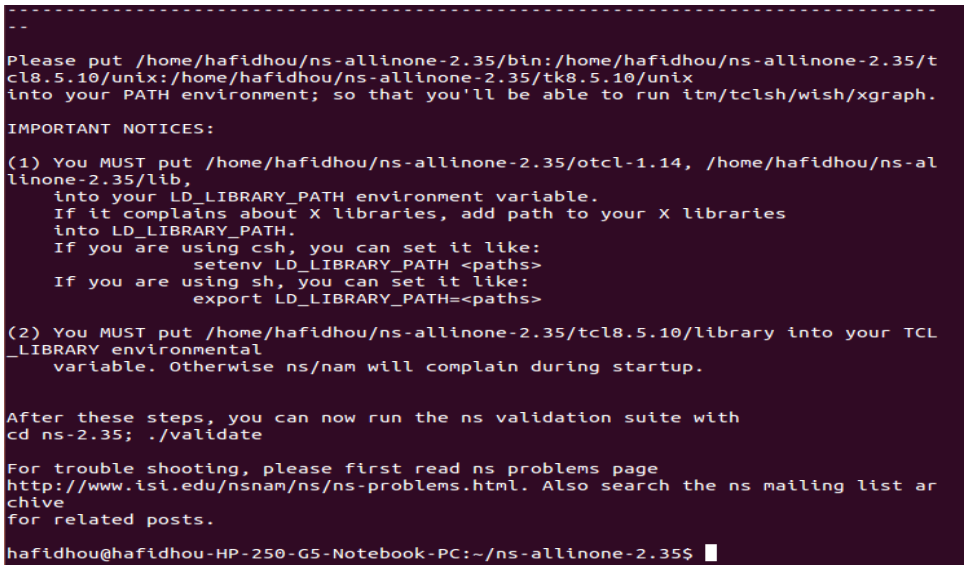


Figure 30: Message sur le terminal après succès de l'installation de NS-2.35

N.B : En cas d'erreurs lors de l'installation, des solutions sont proposées dans [69].

Après l'installation, il faut configurer le path (variables d'environnement) dans le fichier *.bashrc* situé dans le répertoire personnel *~/*. Il suffit de taper la commande qui suit pour ouvrir le fichier *.bashrc* :

```
# gedit ~/.bashrc
```

et ajouté les lignes suivantes à la fin du fichier (en remplaçant */home/hafidhou* par le répertoire dans lequel les fichiers NS-2 ont été extraits) :

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/hafidhou/ns-allinone-2.35/otcl-1.13
NS2_LIB=/home/hafidhou/ns-allinone-2.35/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
GCC=/usr/bin
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB:$G
CC

# TCL_LIBRARY
TCL_LIB=/home/hafidhou/ns-allinone-2.35/tcl8.4.18/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=/home/hafidhou/ns-allinone-2.35/bin:/home/hafidhou/ns-allinone-
2.35/tcl8.4.18/unix:/home/hafidhou/ns-allinone-2.35/tk8.4.18/unix
NS=/home/hafidhou/ns-allinone-2.35/NS-2.35/
NAM=/home/hafidhou/ns-allinone-2.35/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

Après avoir configuré les variables d'environnement, il ne reste qu'à valider les configurations en entrant dans le répertoire */ns-2.35* à partir du répertoire d'installation (ici, le répertoire personnel) et en tapant *validate* comme suit :

```
# cd ~/ns-allinone-2.35/ns-2.35
# ./validate
```

Une fois la validation terminée avec succès, alors NS-2 est bien installé et peut être utilisé. Pour tester, il suffit de taper *ns* sur le terminal et le symbole *%* apparaitra.
Pour tester une simulation, on peut taper ns suivi du nom du fichier tcl sur un terminal :

```
# ns fichier_test.tcl
```

Annexe 2

Installation de VanetMobiSim

Tout d'abord, il faut télécharger le fichier source *VanetMobiSim 1.1* à partir de [70] et s'assurer de disposer d'un **JVM** (**J**ava **V**irtual **M**achine) *Java SDK* version 1.5 ou plus récente (on peut vérifier en tapant la commande *java -version*) et *Apache Ant* sur l'ordinateur.
Après le téléchargement, il faut extraire les fichiers zippés dans un répertoire de base au choix (exemple : le répertoire personnel) et vérifier la présence de :

```
jar/
build.xml
VanetMobiSim-src.jar
VanetMobiSim-samples.jar
mypackages.lst
READ_ME
```

Ensuite, il faut télécharger le fichier source *CanuMobiSim v1.3.4* sur [64] et l'extraire dans le même répertoire que *VanetMobiSim*. Après l'extraction, on doit avoir un sous répertoire nommé « src/ ». Et maintenant le répertoire courant doit contenir :

```
jar/
src/
build.xml
VanetMobiSim-src.jar
VanetMobiSim-samples.jar
mypackages.lst
READ_ME
```

Il reste à exécuter deux commandes pour terminer l'installation de *VanetMobiSim*. Pour se faire, il faut ouvrir un terminal et se placer dans le répertoire de base. Il faut vérifier que Apache Ant est bien configuré et que le fichier build.xml se trouve dans le répertoire de base avant de lancer ant.
On tape la commande :

```
#  ant patch
```

Cette commande permet au programme d'effectuer le patch (corriger et mettre à jour) du répertoire *src/* avec les fichiers sources de *VanetMobiSim*.

En fin, il faut configurer le simulateur et créer les javadocs à l'aide de la commande :

```
#  ant all
```

Les fichiers .jar de VanetMobiSim se trouve dans le sous répertoire *jar/*.

Annexe 3

Fichier *Manhattan.xml*

```
<?xml version="1.0"?>
<!-- Manhattan scenario -->
<!-- 1000x1000 m, 50 nodes, speed: 20-65 km/h, stay: 5-30 s , max
traffic lights: 10, step of traffic lights : 20000 ms, number of
lanes of each road: 2 -->
<universe>
  <dimx>1000.0</dimx>
  <dimy>1000.0</dimy>
  <seed>18</seed>
  <extension
class="de.uni_stuttgart.informatik.canu.mobisim.extensions.NSOutpu
t"/>
  <extension
class="de.uni_stuttgart.informatik.canu.mobisim.simulations.TimeSi
mulation" param="300.0"/>
  <extension
class="de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialM
odel" min_x="0" min_y="0" max_x="1000" max_y="1000">
    <max_traffic_lights>10</max_traffic_lights>
    <number_lane full="true">2</number_lane>
    <reflect_directions>true</reflect_directions>
  </extension>
  <extension name="TrafficLight"
class="eurecom.spatialmodel.extensions.TrafficLight"
step="20000"/>

    <extension class="eurecom.usergraph.UserGraph"
name="userGraph">
     <vertex> <id>0</id> <x>0</x> <y>0</y> </vertex>
     <vertex> <id>1</id> <x>250</x> <y>0</y> </vertex>
     <vertex> <id>2</id> <x>500</x> <y>0</y> </vertex>
     <vertex> <id>3</id> <x>750</x> <y>0</y> </vertex>
     <vertex> <id>4</id> <x>1000</x> <y>0</y> </vertex>

     <vertex> <id>5</id> <x>0</x> <y>250</y> </vertex>
     <vertex> <id>6</id> <x>250</x> <y>250</y> </vertex>
     <vertex> <id>7</id> <x>500</x> <y>250</y> </vertex>
     <vertex> <id>8</id> <x>750</x> <y>250</y> </vertex>
     <vertex> <id>9</id> <x>1000</x> <y>250</y> </vertex>

     <vertex> <id>10</id> <x>0</x> <y>500</y> </vertex>
     <vertex> <id>11</id> <x>250</x> <y>500</y> </vertex>
     <vertex> <id>12</id> <x>500</x> <y>500</y> </vertex>
     <vertex> <id>13</id> <x>750</x> <y>500</y> </vertex>
     <vertex> <id>14</id> <x>1000</x> <y>500</y> </vertex>

     <vertex> <id>15</id> <x>0</x> <y>750</y> </vertex>
     <vertex> <id>16</id> <x>250</x> <y>750</y> </vertex>
     <vertex> <id>17</id> <x>500</x> <y>750</y> </vertex>
     <vertex> <id>18</id> <x>750</x> <y>750</y> </vertex>
     <vertex> <id>19</id> <x>1000</x> <y>750</y> </vertex>
```

```
<vertex> <id>20</id> <x>0</x> <y>1000</y> </vertex>
<vertex> <id>21</id> <x>250</x> <y>1000</y> </vertex>
<vertex> <id>22</id> <x>500</x> <y>1000</y> </vertex>
<vertex> <id>23</id> <x>750</x> <y>1000</y> </vertex>
<vertex> <id>24</id> <x>1000</x> <y>1000</y> </vertex>


<edge> <v1>0</v1> <v2>1</v2> <speed>18.05</speed> </edge>
<edge> <v1>1</v1> <v2>2</v2> <speed>18.05</speed> </edge>
<edge> <v1>2</v1> <v2>3</v2> <speed>18.05</speed> </edge>
<edge> <v1>3</v1> <v2>4</v2> <speed>18.05</speed> </edge>

<edge> <v1>5</v1> <v2>6</v2> <speed>18.05</speed> </edge>
<edge> <v1>6</v1> <v2>7</v2> <speed>18.05</speed> </edge>
<edge> <v1>7</v1> <v2>8</v2> <speed>18.05</speed> </edge>
<edge> <v1>8</v1> <v2>9</v2> <speed>18.05</speed> </edge>

<edge> <v1>10</v1> <v2>11</v2> <speed>18.05</speed> </edge>
<edge> <v1>11</v1> <v2>12</v2> <speed>18.05</speed> </edge>
<edge> <v1>12</v1> <v2>13</v2> <speed>18.05</speed> </edge>
<edge> <v1>13</v1> <v2>14</v2> <speed>18.05</speed> </edge>

<edge> <v1>15</v1> <v2>16</v2> <speed>18.05</speed> </edge>
<edge> <v1>16</v1> <v2>17</v2> <speed>18.05</speed> </edge>
<edge> <v1>17</v1> <v2>18</v2> <speed>18.05</speed> </edge>
<edge> <v1>18</v1> <v2>19</v2> <speed>18.05</speed> </edge>

<edge> <v1>20</v1> <v2>21</v2> <speed>18.05</speed> </edge>
<edge> <v1>21</v1> <v2>22</v2> <speed>18.05</speed> </edge>
<edge> <v1>22</v1> <v2>23</v2> <speed>18.05</speed> </edge>
<edge> <v1>23</v1> <v2>24</v2> <speed>18.05</speed> </edge>


<edge> <v1>0</v1> <v2>5</v2> <speed>18.05</speed> </edge>
<edge> <v1>5</v1> <v2>10</v2> <speed>18.05</speed> </edge>
<edge> <v1>10</v1> <v2>15</v2> <speed>18.05</speed> </edge>
<edge> <v1>15</v1> <v2>20</v2> <speed>18.05</speed> </edge>

<edge> <v1>1</v1> <v2>6</v2> <speed>18.05</speed> </edge>
<edge> <v1>6</v1> <v2>11</v2> <speed>18.05</speed> </edge>
<edge> <v1>11</v1> <v2>16</v2> <speed>18.05</speed> </edge>
<edge> <v1>16</v1> <v2>21</v2> <speed>18.05</speed> </edge>

<edge> <v1>2</v1> <v2>7</v2> <speed>18.05</speed> </edge>
<edge> <v1>7</v1> <v2>12</v2> <speed>18.05</speed> </edge>
<edge> <v1>12</v1> <v2>17</v2> <speed>18.05</speed> </edge>
<edge> <v1>17</v1> <v2>22</v2> <speed>18.05</speed> </edge>

<edge> <v1>3</v1> <v2>8</v2> <speed>18.05</speed> </edge>
<edge> <v1>8</v1> <v2>13</v2> <speed>18.05</speed> </edge>
<edge> <v1>13</v1> <v2>18</v2> <speed>18.05</speed> </edge>
<edge> <v1>18</v1> <v2>23</v2> <speed>18.05</speed> </edge>

<edge> <v1>4</v1> <v2>9</v2> <speed>18.05</speed> </edge>
<edge> <v1>9</v1> <v2>14</v2> <speed>18.05</speed> </edge>
```

```
      <edge> <v1>14</v1> <v2>19</v2> <speed>18.05</speed> </edge>
      <edge> <v1>19</v1> <v2>24</v2> <speed>18.05</speed> </edge>

  </extension>



<extension name="PosGen"
class="de.uni_stuttgart.informatik.canu.tripmodel.generators.Rando
mInitialPositionGenerator"/>
  <extension name="TripGen"
class="de.uni_stuttgart.informatik.canu.tripmodel.generators.Rando
mTripGenerator">
    <reflect_directions>true</reflect_directions>
       <minstay>5.0</minstay> <maxstay>30.0</maxstay>
  </extension>
  <nodegroup n="50">
    <extension class="polito.uomm.IDM_LC"
initposgenerator="PosGen" tripgenerator="TripGen">
      <minspeed>5.55</minspeed>
      <maxspeed>18.05</maxspeed>
      <step>0.1</step>
     <b>0.5</b>
    </extension>
  </nodegroup>


     <extension
class="de.uni_stuttgart.informatik.canu.mobisimadd.extensions.GUI"
>
       <width>640</width>
          <height>480</height>
          <step>1</step>
     </extension>
</universe>
```

Annexe 4

Fichier *Urbain.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Cars in a City Center using the SpaceGraph Traffic Light. -->
<universe>
  <dimx>1000.0</dimx>
  <dimy>1000.0</dimy>
  <seed>1</seed>
  <extension
class="de.uni_stuttgart.informatik.canu.mobisim.extensions.NSOutpu
t"/>
  <extension
class="de.uni_stuttgart.informatik.canu.mobisim.simulations.TimeSi
mulation" param="300.0"/>
      <extension name="NewSpatialModel"
class="de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialM
odel" traffic_light="NewTrafficLight" min_x="0" max_x="1000"
min_y="0" max_y="1000">
        <max_traffic_lights>6</max_traffic_lights>
        <reflect_directions>true</reflect_directions>
           <number_lane full="false" max="4"
dir="true">2</number_lane>
      </extension>
      <extension name="NewTrafficLight"
class="eurecom.spatialmodel.extensions.TrafficLight"
spatial_model="NewSpatialModel" step="10000"/>
 <extension class="eurecom.spacegraph.SpaceGraph"
spatial_model="NewSpatialModel" traffic_light="NewTrafficLight"
cluster="true">
    <clusters density="0.000004">
            <cluster id="downtown">
                <density>0.0002</density>
                    <ratio>0.1</ratio>
                    <speed>16.66</speed>
               </cluster>
               <cluster id="residential">
               <density>0.00005</density>
                    <ratio>0.4</ratio>
                    <speed>11.11</speed>
               </cluster>
               <cluster id="suburban">
                <density>0.00001</density>
                    <ratio>0.5</ratio>
                    <speed>27.77</speed>
               </cluster>
          </clusters>
 </extension>
  <extension name="PosGen"
class="de.uni_stuttgart.informatik.canu.tripmodel.generators.Rando
mInitialPositionGenerator" spatial_model="NewSpatialModel"/>
  <extension name="TripGen"
class="de.uni_stuttgart.informatik.canu.tripmodel.generators.Rando
mTripGenerator" spatial_model="NewSpatialModel">
    <reflect_directions>true</reflect_directions>
```

```
        <minstay>5.0</minstay> <maxstay>30.0</maxstay>
  </extension>
  <nodegroup n="100">
    <extension class="polito.uomm.IDM_LC"
spatial_model="NewSpatialModel" initposgenerator="PosGen"
tripgenerator="TripGen">
      <minspeed>5.55</minspeed>
      <maxspeed>27.77</maxspeed>
      <step>0.1</step>
     <b>0.5</b>
    </extension>
  </nodegroup>
  <extension
class="de.uni_stuttgart.informatik.canu.mobisimadd.extensions.GUI"
spatial_model="NewSpatialModel">
    <width>640</width>
    <height>480</height>
    <step>1</step>
  </extension>
<!--
  <extension
class="de.uni_stuttgart.informatik.canu.spatialmodel.extensions.Du
mpSpatialModel" spatial_model="NewSpatialModel"
output="dumped_graph.fig"/>
-->
</universe>
```

Annexe 5

Script *rp-AODV.tcl*

```
#===================================
#     Simulation parameters setup
#===================================

#check the parameters
if {$argc != 7} {
      puts "Invalid options"
      puts "Usage: "
      puts "ns rp.tcl routing_protocols node_num max_connection
sending_rate pause_time max_speed sim_time"
      exit
}

#get
set rp           [lindex $argv 0]
set nn           [lindex $argv 1]
set co           [lindex $argv 2]
set ra           [lindex $argv 3]
set p      [lindex $argv 4]
set v      [lindex $argv 5]
set sim          [lindex $argv 6]
#options for the mobile-nodes
#=================================================================
set opt(chan)         Channel/WirelessChannel
set opt(prop)         Propagation/TwoRayGround
;#Propagation/Shadowing

#--------------------for IEEE 802.11 - using on MANETs------------

#set opt(netif)       Phy/WirelessPhy
#set opt(mac)         Mac/802_11

#-----------------------------------------------------------------

### Added by Hafidhou Ibrahim Ahmed Said - Sciences and
Technologies Faculty of Fez, Morocco ###

#--------------------for IEEE 802.11p - using on VANETs-----------
set opt(netif)          Phy/WirelessPhyExt          ;# network
interface type
set opt(mac)            Mac/802_11Ext                 ;# MAC type

Phy/WirelessPhyExt set CSThresh_                 3.162e-12   ;#-85
dBm Wireless interface sensitivity (sensitivity defined in the
standard)
Phy/WirelessPhyExt set Pt_                       0.001
Phy/WirelessPhyExt set freq_                     5.9e+9
Phy/WirelessPhyExt set noise_floor_              1.26e-13    ;#-99
dBm for 10MHz bandwidth
Phy/WirelessPhyExt set L_                        1.0
;#default radio circuit gain/loss
```

```
Phy/WirelessPhyExt set PowerMonitorThresh_      6.310e-14   ;#-
102dBm power monitor  sensitivity
Phy/WirelessPhyExt set HeaderDuration_          0.000040    ;#40
us
Phy/WirelessPhyExt set BasicModulationScheme_   0
Phy/WirelessPhyExt set PreambleCaptureSwitch_   1
Phy/WirelessPhyExt set DataCaptureSwitch_       0
Phy/WirelessPhyExt set SINR_PreambleCapture_    2.5118;     ;# 4
dB
Phy/WirelessPhyExt set SINR_DataCapture_        100.0;      ;# 10
dB
Phy/WirelessPhyExt set trace_dist_              1e6         ;# PHY
trace until distance of 1 Mio. km ("infinty")
Phy/WirelessPhyExt set PHY_DBG_                 0

Mac/802_11Ext set CWMin_                        15
Mac/802_11Ext set CWMax_                        1023
Mac/802_11Ext set SlotTime_                     0.000013
Mac/802_11Ext set SIFS_                         0.000032
Mac/802_11Ext set ShortRetryLimit_              7
Mac/802_11Ext set LongRetryLimit_               4
Mac/802_11Ext set HeaderDuration_               0.000040
Mac/802_11Ext set SymbolDuration_               0.000008
Mac/802_11Ext set BasicModulationScheme_        0
Mac/802_11Ext set use_802_11a_flag_             true
Mac/802_11Ext set RTSThreshold_                 2346
Mac/802_11Ext set MAC_DBG                       0

#---------------------------------------------------------------------
#####################################################################

if {$rp=="DSR"} {
      set opt(ifq)    CMUPriQueue ;#for DSR routing protocol
} else {
set opt(ifq)          Queue/DropTail/PriQueue ;#for DSDV and AODV
}
set opt(ll)           LL
set opt(ant)               Antenna/OmniAntenna

set opt(x)      1001 ;#the simulation scenario is 1001x1001
set opt(y)      1001
set opt(cp)           "./trafic/cbr-$nn-$co-$ra.tcl"
set opt(sc)           "./scen/scen-$nn.tcl"
#set opt(sc)          "./scen/scen-$nn-urbain.tcl"

set opt(ifqlen)       50
set opt(nn)           $nn   ;#num of mobile nodes
#set opt(seed)        1.0   ;#the seed of random num
set opt(stop)         $sim  ;#the end time
set opt(tr)           "$rp/$rp-$nn-$co-$ra-$p-$v-$sim-
$opt(x)x$opt(y).tr"   ;#trace file
set opt(nam)          "nam/$rp-$nn-$co-$ra-$p-$v-$sim-
$opt(x)x$opt(y).nam"  ;#nam trace file
set opt(rp)              $rp      ;#routing protocol
```

```
#Main Program

set ns_ [new Simulator]

$ns_ use-newtrace

set tracefd [open $opt(tr) w]
$ns_ trace-all $tracefd

set namtracefd [open $opt(nam) w]
$ns_ namtrace-all-wireless $namtracefd $opt(x) $opt(y)

set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)

set god_ [create-god $opt(nn)]

set chan [new $opt(chan)]

#===================================
#     Mobile node parameter setup
#===================================
$ns_ node-config -adhocRouting  $opt(rp) \
                 -llType        $opt(ll) \
                 -macType       $opt(mac) \
                 -ifqType       $opt(ifq) \
                 -ifqLen        $opt(ifqlen) \
                 -antType       $opt(ant) \
                 -propType      $opt(prop) \
                 -phyType       $opt(netif) \
                 -channel       $chan \
                 -topoInstance  $topo \
                 -agentTrace    ON \
                 -routerTrace   ON \
                 -macTrace      OFF \
                 -movementTrace ON

for {set i 0} {$i < $opt(nn) } {incr i} {
      set node_($i) [$ns_ node]
      $node_($i) random-motion 0
}

#set node_(31) [$ns_ node]

puts "loading traffic file: $opt(cp)"
source $opt(cp)

puts "loading scene file: $opt(sc)"
source $opt(sc)

### Added for blackhole attack - Hafidhou Ibrahim Ahmed Said ###
############ Blackhole nodes definition ############
```

```
#---Colors and labels of blackhole nodes---
$node_(31) color red
$ns_ at 0.0 "$node_(31) color red"
$ns_ at 0.0 "$node_(31) label Attacker"

$node_(33) color red
$ns_ at 0.0 "$node_(33) color red"
$ns_ at 0.0 "$node_(33) label Attacker"

$node_(37) color red
$ns_ at 0.0 "$node_(37) color red"
$ns_ at 0.0 "$node_(37) label Attacker"

$node_(39) color red
$ns_ at 0.0 "$node_(39) color red"
$ns_ at 0.0 "$node_(39) label Attacker"

$node_(48) color red
$ns_ at 0.0 "$node_(48) color red"
$ns_ at 0.0 "$node_(48) label Attacker"
#-----------------------------------------

#----------Blackhole nodes definition--------
$ns_ at 0.0 "[$node_(31) set ragent_] hacker"
$ns_ at 0.0 "[$node_(33) set ragent_] hacker"
$ns_ at 0.0 "[$node_(37) set ragent_] hacker"
$ns_ at 0.0 "[$node_(39) set ragent_] hacker"
$ns_ at 0.0 "[$node_(48) set ragent_] hacker"
#-----------------------------------------
#####################################################################

for {set i 0} {$i < $opt(nn)} {incr i} {
     $ns_ initial_node_pos $node_($i) 10
}

for {set i 0} {$i < $opt(nn)} {incr i} {
     $ns_ at $opt(stop).000000001 "$node_($i) reset"
}

$ns_ at $opt(stop).000000001 "finish"

proc finish {} {
     global ns_ tracefd namtracefd
     global ns_ tracefd
     $ns_ flush-trace
     close $tracefd
     close $namtracefd
     #exec nam ./nam/AODV-10-15-8-20-20-300-1001x1001.nam &
     exit 0
}

proc timeReport {interval} {
     global ns_ tracefd namtracefd
     global ns_ tracefd
```

```
  set now [$ns_ now]
  #puts "Time=[clock format [clock second] -format "%H:%M"](min),
Sim=[format %.1f $now](sec)"

  flush $tracefd
  flush $namtracefd

  $ns_ at [expr $now+$interval] "timeReport $interval"
}

$ns_ at 0 "timeReport 1"

$ns_ run
```

Annexe 6

Script *rp-GPSR.tcl*

```
# ======================================================================
# Default Script Options
# ======================================================================
set opt(chan)         Channel/WirelessChannel
set opt(prop)         Propagation/TwoRayGround
set opt(netif)        Phy/WirelessPhy
set opt(mac)          Mac/802_11
#set opt(netif)       Phy/WirelessPhyExt
#set opt(mac)         Mac/802_11Ext

     ### Added by Hafidhou Ibrahim Ahmed Said ###
#--------------------for IEEE 802.11p----------------------
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0

# The transimssion radio range
Phy/WirelessPhy set Pt_ 0.2818       ;# 250m
#-----------------------------------------
############################################################

LL set mindelay_      50us
LL set delay_              25us
LL set bandwidth_          0     ;# not used

Agent/Null set sport_      0
Agent/Null set dport_      0

Agent/CBR set sport_       0
Agent/CBR set dport_       0

Agent/TCPSink set sport_   0
Agent/TCPSink set dport_   0

Agent/TCP set sport_       0
Agent/TCP set dport_       0
Agent/TCP set packetSize_  1460

Queue/DropTail/PriQueue set Prefer_Routing_Protocols    1

set opt(ifq)    Queue/DropTail/PriQueue ;#for DSDV and AODV
```

```
set opt(ll)           LL
set opt(ant)            Antenna/OmniAntenna

set opt(x)      1934 ;#the simulation scenario is 1001x1001
set opt(y)      100
#set opt(cp)          "./trafic/GPSR-cbr-25.tcl"
#set opt(sc)          "./scen/scen-25-grille.tcl"

set opt(ifqlen)       50
set opt(nn)           25;#num of mobile nodes
#set opt(seed)        1.0;#the seed of random num
set opt(stop)         50;#the end time
set opt(tr)           "GPSR/GPSR-25.tr";#trace file
set opt(nam)          "nam/GPSR-25.nam"    ;#nam trace file
set opt(rp)             GPSR;#routing protocol

# ======================================================================

# Agent/GPSR setting
Agent/GPSR set planar_type_  1   ;#1=GG planarize, 0=RNG
planarize
Agent/GPSR set hello_period_   2.0 ;#Hello message period

# ======================================================================

proc usage { argv0 }  {
      puts "Usage: $argv0"
      puts "\tmandatory arguments:"
      puts "\t\t\[-x MAXX\] \[-y MAXY\]"
      puts "\toptional arguments:"
      puts "\t\t\[-cp conn pattern\] \[-sc scenario\] \[-nn
nodes\]"
      puts "\t\t\[-seed seed\] \[-stop sec\] \[-tr tracefile\]\n"
}

proc getopt {argc argv} {
      global opt
      lappend optlist cp nn seed sc stop tr x y

      for {set i 0} {$i < $argc} {incr i} {
            set arg [lindex $argv $i]
            if {[string range $arg 0 0] != "-"} continue

            set name [string range $arg 1 end]
            set opt($name) [lindex $argv [expr $i+1]]
      }
}

proc log-movement {} {
    global logtimer ns_ ns

    set ns $ns_
    source ~/gpsr-keliu/ns-allinone-2.35/ns-
2.35/tcl/mobility/timer.tcl
    Class LogTimer -superclass Timer
```

```
    LogTimer instproc timeout {} {
      global opt node_;
      for {set i 0} {$i < $opt(nn)} {incr i} {
          $node_($i) log-movement
      }
      $self sched 0.1
    }

    set logtimer [new LogTimer]
    $logtimer sched 0.1
}
#
# ===================================================================
# Main Program
# ===================================================================
#
source ~/gpsr-keliu/ns-allinone-2.35/ns-2.35/tcl/lib/ns-
bsnode.tcl
source  ~/gpsr-keliu/ns-allinone-2.35/ns-
2.35/tcl/mobility/com.tcl

# do the get opt again incase the routing protocol file added
some more
# options to look for
getopt $argc $argv

if { $opt(x) == 0 || $opt(y) == 0 } {
      usage $argv0
      exit 1
}

#
# Initialize Global Variables
#
set ns_          [new Simulator]

set chan   [new $opt(chan)]
set prop   [new $opt(prop)]
set topo   [new Topography]

set tracefd   [open $opt(tr) w]
$ns_ trace-all  $tracefd

set namfile [open $opt(nam) w]
$ns_ namtrace-all $namfile

$topo load_flatgrid $opt(x) $opt(y)
$prop topography $topo

#
# Create God
#
set god_ [create-god $opt(nn)]

#
```

```
# Create the specified number of nodes $opt(nn) and "attach" them
the channel.
# Each routing protocol script is expected to have defined a proc
# create-mobile-node that builds a mobile node and inserts it
into the
# array global $node_($i)
#
$ns_ node-config -adhocRouting $opt(rp) \
                 -llType $opt(ll) \
                 -macType $opt(mac) \
                 -ifqType $opt(ifq) \
                 -ifqLen $opt(ifqlen) \
                 -antType $opt(ant) \
                 -propType $opt(prop) \
                 -phyType $opt(netif) \
                 -channelType $opt(chan) \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace ON \
                 -movementTrace ON

source ./gpsr.tcl

for {set i 0} {$i < $opt(nn) } {incr i} {
   gpsr-create-mobile-node $i
}

for {set i 0} {$i < $opt(nn) } {incr i} {
     set node_($i) [$ns_ node]
     $node_($i) random-motion 0
}

# Source the Connection and Movement scripts

#if { $opt(cp) == "" } {
#    puts "*** NOTE: no connection pattern specified."
#        set opt(cp) "none"
#} else {
#    puts "Loading connection pattern..."
#    source $opt(cp)
#}

#
# Tell all the nodes when the simulation ends
#
for {set i 0} {$i < $opt(nn) } {incr i} {
    $ns_ at $opt(stop).000000001 "$node_($i) reset";
}
$ns_ at $opt(stop).00000001 "puts \"NS EXITING...\" ; $ns_ halt"

#if { $opt(sc) == "" } {
#    puts "*** NOTE: no scenario file specified."
#        set opt(sc) "none"
#} else {
```

```
#     puts "Loading scenario file..."
#     source $opt(sc)
#     puts "Load complete..."
#}

### Added for blackhole attack - Hafidhou Ibrahim Ahmed Said ###
     ### Blackhole nodes definition ###

#---Colors and labels of blackhole nodes---
$node_(6) color red
$ns_ at 0.0 "$node_(6) color red"
$ns_ at 0.0 "$node_(6) label Attacker"

$node_(7) color red
$ns_ at 0.0 "$node_(7) color red"
$ns_ at 0.0 "$node_(7) label Attacker"

$node_(12) color red
$ns_ at 0.0 "$node_(12) color red"
$ns_ at 0.0 "$node_(12) label Attacker"

#----------Blackhole nodes definition--------

$ns_ at 0.0 "[$node_(6) set ragent_] hacker"
$ns_ at 0.0 "[$node_(7) set ragent_] hacker"
$ns_ at 0.0 "[$node_(12) set ragent_] hacker"

#-----------------------------------------
##############################################################

#===================================
#        Nodes Definition
#===================================
#Create 25 nodes

$node_(0) set X_ 301
$node_(0) set Y_ 499
$node_(0) set Z_ 0.0
$ns_ initial_node_pos $node_(0) 20
$node_(1) set X_ 501
$node_(1) set Y_ 499
$node_(1) set Z_ 0.0
$ns_ initial_node_pos $node_(1) 20
$node_(2) set X_ 701
$node_(2) set Y_ 499
$node_(2) set Z_ 0.0
$ns_ initial_node_pos $node_(2) 20
$node_(3) set X_ 901
$node_(3) set Y_ 499
$node_(3) set Z_ 0.0
$ns_ initial_node_pos $node_(3) 20
$node_(4) set X_ 1101
$node_(4) set Y_ 499
$node_(4) set Z_ 0.0
$ns_ initial_node_pos $node_(4) 20
```

```
$node_(5) set X_ 301
$node_(5) set Y_ 299
$node_(5) set Z_ 0.0
$ns_ initial_node_pos $node_(5) 20
$node_(6) set X_ 501
$node_(6) set Y_ 299
$node_(6) set Z_ 0.0
$ns_ initial_node_pos $node_(6) 20
$node_(7) set X_ 701
$node_(7) set Y_ 299
$node_(7) set Z_ 0.0
$ns_ initial_node_pos $node_(7) 20
$node_(8) set X_ 901
$node_(8) set Y_ 299
$node_(8) set Z_ 0.0
$ns_ initial_node_pos $node_(8) 20
$node_(9) set X_ 1101
$node_(9) set Y_ 299
$node_(9) set Z_ 0.0
$ns_ initial_node_pos $node_(9) 20
$node_(10) set X_ 301
$node_(10) set Y_ 99
$node_(10) set Z_ 0.0
$ns_ initial_node_pos $node_(10) 20
$node_(11) set X_ 501
$node_(11) set Y_ 99
$node_(11) set Z_ 0.0
$ns_ initial_node_pos $node_(11) 20
$node_(12) set X_ 701
$node_(12) set Y_ 99
$node_(12) set Z_ 0.0
$ns_ initial_node_pos $node_(12) 20
$node_(13) set X_ 901
$node_(13) set Y_ 99
$node_(13) set Z_ 0.0
$ns_ initial_node_pos $node_(13) 20
$node_(14) set X_ 1101
$node_(14) set Y_ 99
$node_(14) set Z_ 0.0
$ns_ initial_node_pos $node_(14) 20
$node_(15) set X_ 301
$node_(15) set Y_ -101
$node_(15) set Z_ 0.0
$ns_ initial_node_pos $node_(15) 20
$node_(16) set X_ 501
$node_(16) set Y_ -101
$node_(16) set Z_ 0.0
$ns_ initial_node_pos $node_(16) 20
$node_(17) set X_ 701
$node_(17) set Y_ -101
$node_(17) set Z_ 0.0
$ns_ initial_node_pos $node_(17) 20
$node_(18) set X_ 901
$node_(18) set Y_ -101
$node_(18) set Z_ 0.0
```

```
$ns_ initial_node_pos $node_(18) 20
$node_(19) set X_ 1101
$node_(19) set Y_ -101
$node_(19) set Z_ 0.0
$ns_ initial_node_pos $node_(19) 20
$node_(20) set X_ 301
$node_(20) set Y_ -301
$node_(20) set Z_ 0.0
$ns_ initial_node_pos $node_(20) 20
$node_(21) set X_ 501
$node_(21) set Y_ -301
$node_(21) set Z_ 0.0
$ns_ initial_node_pos $node_(21) 20
$node_(22) set X_ 701
$node_(22) set Y_ -301
$node_(22) set Z_ 0.0
$ns_ initial_node_pos $node_(22) 20
$node_(23) set X_ 901
$node_(23) set Y_ -301
$node_(23) set Z_ 0.0
$ns_ initial_node_pos $node_(23) 20
$node_(24) set X_ 1101
$node_(24) set Y_ -301
$node_(24) set Z_ 0.0
$ns_ initial_node_pos $node_(24) 20

#===================================
#        Agents Definition
#===================================
#Setup a UDP connection
set udp0 [new Agent/UDP]
$ns_ attach-agent $node_(0) $udp0
set null1 [new Agent/Null]
$ns_ attach-agent $node_(4) $null1
$ns_ connect $udp0 $null1
$udp0 set packetSize_ 1500

#Setup a UDP connection
set udp2 [new Agent/UDP]
$ns_ attach-agent $node_(5) $udp2
set null3 [new Agent/Null]
$ns_ attach-agent $node_(4) $null3
$ns_ connect $udp2 $null3
$udp2 set packetSize_ 1500

#Setup a UDP connection
set udp4 [new Agent/UDP]
$ns_ attach-agent $node_(10) $udp4
set null5 [new Agent/Null]
$ns_ attach-agent $node_(4) $null5
$ns_ connect $udp4 $null5
$udp4 set packetSize_ 1500

#Setup a UDP connection
set udp6 [new Agent/UDP]
```

```
$ns_ attach-agent $node_(15) $udp6
set null7 [new Agent/Null]
$ns_ attach-agent $node_(4) $null7
$ns_ connect $udp6 $null7
$udp6 set packetSize_ 1500

#Setup a UDP connection
set udp8 [new Agent/UDP]
$ns_ attach-agent $node_(20) $udp8
set null9 [new Agent/Null]
$ns_ attach-agent $node_(4) $null9
$ns_ connect $udp8 $null9
$udp8 set packetSize_ 1500

#===================================
#        Applications Definition
#===================================
#Setup a CBR Application over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp0
$cbr1 set packetSize_ 1000
$cbr1 set rate_ 1.0Mb
$cbr1 set random_ null
$ns_ at 1.0 "$cbr1 start"
$ns_ at 50.0 "$cbr1 stop"

#Setup a CBR Application over UDP connection
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp2
$cbr2 set packetSize_ 1000
$cbr2 set rate_ 1.0Mb
$cbr2 set random_ null
$ns_ at 1.0 "$cbr2 start"
$ns_ at 50.0 "$cbr2 stop"

#Setup a CBR Application over UDP connection
set cbr3 [new Application/Traffic/CBR]
$cbr3 attach-agent $udp4
$cbr3 set packetSize_ 1000
$cbr3 set rate_ 1.0Mb
$cbr3 set random_ null
$ns_ at 1.0 "$cbr3 start"
$ns_ at 50.0 "$cbr3 stop"

#Setup a CBR Application over UDP connection
set cbr4 [new Application/Traffic/CBR]
$cbr4 attach-agent $udp6
$cbr4 set packetSize_ 1000
$cbr4 set rate_ 1.0Mb
$cbr4 set random_ null
$ns_ at 1.0 "$cbr4 start"
$ns_ at 50.0 "$cbr4 stop"

#Setup a CBR Application over UDP connection
set cbr5 [new Application/Traffic/CBR]
```

```
$cbr5 attach-agent $udp8
$cbr5 set packetSize_ 1000
$cbr5 set rate_ 1.0Mb
$cbr5 set random_ null
$ns_ at 1.0 "$cbr5 start"
$ns_ at 50.0 "$cbr5 stop"

puts "Starting Simulation..."

proc finish {} {
    #global ns_ tracefd namfile
    global ns_ tracefd
    $ns_ flush-trace
    close $tracefd
    #close $namfile
    #exec nam "nam/GPSR-25.nam"
    exit 0
}

$ns_ at $opt(stop) "finish"

$ns_ run
```

Annexe 7

Fichier *perf.awk*

```
BEGIN {

  totalreceved = 0;
  PDR = 0.0;
  totalsend = 0;
  total_time = 0.0;
  nb_Pacquet_routage=0.0;
  cout_routage=0.0;
  nb_Pacquet_donnee=0.0;
  totalsend_data=0.0;
  gigue=0.0;
  highest_packet_id = 0;
  jitter_sum=0.0;
  last_delay=0.0;
  n=0.0;
  MAX_NODES = 1000;
  flow = 0;
  delay = 0.0;

 }
{
  action = $1;
  time = $3;
  layer = $19;
  pkttype = $35;
  packet_id = $41;
  msg=$51;

  if (packet_id > highest_packet_id )
         highest_packet_id = packet_id ;
  if ( action == "s" && layer == "AGT")
         if ( start_time[packet_id] == 0 ) start_time[packet_id] = time;
  if (action != "d" ){
         if (action == "r" && layer == "AGT" && pkttype=="cbr") {
           if( end_time[packet_id] == 0 ) {
                split ($31,a,".");
                split ($33,b,".");
                src_id[packet_id] = a[1];
                dst_id[packet_id] = b[1];
           }
              end_time[packet_id] = time;
        }
  }
  else
        end_time[packet_id] = -1;

  if ((layer == "AGT") &&(action == "r") && (pkttype=="cbr")) totalreceved
++;
  if ((layer == "AGT") &&(action == "s") && (pkttype=="cbr")) totalsend
++;
  #if ((layer == "RTR") && (action == "s")) nb_Pacquet_routage++;
  if (layer == "AGT") nb_Pacquet_donnee++;
```

```
  if (layer=="RTR") {
     if ( (action=="s" || action=="f") ) {
         if (pkttype == "AODV" || pkttype == "DSR" || pkttype == "message"
|| pkttype=="OLSR") {#message pour DSDV
                nb_Pacquet_routage++;
          }
         else (pkttype=="cbr") totalsend_data ++;
     }
   }

}

END  {
     for ( packet_id = 0; packet_id <= highest_packet_id; packet_id++ ){
       start = start_time[packet_id];
       end = end_time[packet_id];
       end_to_end = end - start;
       if ( start < end )
              total_time = total_time + end_to_end;
     }
    for ( i = 0; i <= highest_packet_id; i++ ) {

          if( end_time[i] <= 0.0 )
               continue;

          delay = (end_time[i] - start_time[i]);
          flow = src_id[i] * MAX_NODES + dst_id[i];
          last_delay = last_delay_by_flow[flow];
          if( last_delay > 0.0){
               if ( delay > last_delay) jitter_sum += delay - last_delay;
               else jitter_sum +=  last_delay - delay;

          }
               n++;
               last_delay_by_flow[flow] = delay;

     }

    delay = total_time/totalreceved;
    PDR  = (totalreceved/totalsend)*100;
    gigue=jitter_sum/n;
    cout_routage=nb_Pacquet_routage/totalreceved;
    efficacite=totalsend_data/(nb_Pacquet_routage+totalsend_data);
    printf("%f %f %f %f %f\n",PDR, delay, gigue, cout_routage,efficacite);
  }
```