Misbehaviors detection schemes in mobile ad hoc networks

Mohammad Rmayti

Thèse
de doctorat
de l'UTT

Mohammad RMAYTI

Misbehaviors Detection Schemes in Mobile Ad Hoc Networks

Spécialité :
Ingénierie Sociotechnique des Connaissances, des Réseaux et du Développement Durable

2016TROY0029 **Année 2016**

utt
université de technologie
Troyes

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : INGENIERIE SOCIOTECHNIQUE DES CONNAISSANCES, DES RESEAUX ET DU DEVELOPPEMENT DURABLE

présentée et soutenue par

Mohammad RMAYTI

le 30 septembre 2016

Misbehaviors Detection Schemes in Mobile Ad hoc Networks

JURY

M. B. COUSIN	PROFESSEUR DES UNIVERSITES	Président
M. A. BOUABDALLAH	PROFESSEUR DES UNIVERSITES	Rapporteur
Mme D. GAÏTI	PROFESSEUR DES UNIVERSITES	Directrice de thèse
M. R. KHATOUN	MAITRE DE CONFERENCES	Directeur de thèse
M. P. LORENZ	PROFESSEUR DES UNIVERSITES	Rapporteur
M. S. ZEADALLY	ASSOCIATE PROFESSOR	Examinateur

Acknowledgments

This thesis appears in its current form due to the assistance and guidance of several people who, I believe, deserve this honor more than myself. I feel relieve to have the opportunity to express my gratitude for all of them.
I am greatly indebted to my supervisors M. Rida Khatoun and Mme. Dominique Gaiti for their excellent advice, support and encouragement they had given me over the last three years.
During my thesis, I really enjoyed working with M. Lyes Khoukhi, and I hope that we will continue working together. He spent a lot of time and efforts to enhance my skills as a researcher, especially in writing articles.
I would like to express my sincere thankfulness to M. Youcef Begriche, who has been keeping a close eye on the progress of my work from the very beginning of this thesis.
I wish equally to express my gratitude without reservation to the CIOES association, who supported me since my B.S and provided funding for my PhD studies in France.
I would like to thank all members of the ERA laboratory and INFRES department.
I am thankful to all my Lebanese friends in France, especially in Troyes, who always give me love, respect and support.
I would like to thank all my friends and colleagues in Lebanon who encouraged me during the past years, especially Eng. Hussein Awada and Eng. Hassan Rmayti.
I am very thankful to Al-Zaïm Hassan Rmayti, Mr. Mohammad Rmayti, Mr. Imad Rmayti and Mr. Zaki Rmayti who have always loved me, encouraged me and supported me.
I am deeply indebted to my Father *Ahmad* and my Mother *Rabab* who have encouraged me to continue my studies and overcome all the challenges. They are always the basic reason behind all the successes that I did and I would do.
I thank my Queens, my dear sisters, Fatima, Zainab and Batoul, and my little hero, my dear brother Ali. I dedicate this thesis for my wonderful Grandfather Aref and my Uncle Kamal who almost died while I was doing my PhD.

Dedicated to my dear parents, Ahmad & Rabab, with love and respect.

"The apple never falls far from the tree."

Proverb

Contents

Résumé

Introduction générale

Contexte et motivation

Au cours de la dernière décennie, le monde a témoigné des progrès importants en informatique mobile ainsi que le développement des technologies sans fil, ce qui a permis aux utilisateurs d'accéder à leurs services et leurs informations d'une manière omniprésente. En outre, les technologies de communication mobile ont joué un rôle important dans l'amélioration de la qualité de vie humaine dans le contexte de villes intelligentes (Smart city) [1].
Aujourd'hui, les équipements mobiles sans fil sont indispensables dans la vie quotidienne des millions d'utilisateurs, qui utilisent en permanence leurs ordinateurs portables, leurs tablettes et leurs smartphones pour bénéficier de plusieurs types de services informatiques. Ces équipements, même s'ils ont des caractéristiques hétérogènes, ils peuvent partager leurs ressources et constituent un réseau mobile ad hoc ou MANET (Mobile Ad hoc Network).
Un réseau MANET est une collection d'*entités* ou *nœuds* mobiles qui peuvent se connecter d'une manière dynamique pour constituer un réseau ayant une topologie arbitraire et temporaire [2]. Les réseaux MANET ne s'appuient sur aucune infrastructure existante, où les nœuds participent d'une manière coopérative pour assurer le routage des messages de données et de contrôle sans le besoin d'une administration centrale. Par conséquent, chaque nœud du réseau joue le rôle d'un point terminal et d'un routeur en même temps. Grâce à ses caractéristiques, les réseaux MANET sont largement utilisés pour assurer la communication dans les situations où il est difficile de déployer une infrastructure réseau, comme les champs de bataille et les opérations de secours.
Malgré les facilités offertes par les réseaux MANET, ils sont vulnérables aux plusieurs attaques de sécurité. En fait, les nœuds qui constituent un réseau MANET proviennent souvent des places différentes et ne se connaissent pas à l'avance. Dans tel contexte exigeant, les nœuds du réseau sont fondamentalement responsables d'assurer les différents services du réseau d'une manière coopérative et totalement décentralisée.

Cependant, les protocoles de routage ad hoc existants ne supportent pas la réalisation de ces responsabilités sous les meilleures conditions de sécurité, surtout que la plupart de ces protocoles assument que les nœuds du réseau sont dignes de confiance. Par conséquent, plusieurs types d'attaques peuvent être réalisées contre les différentes couches du réseau, dont la couche de routage. Par exemple, un nœud malveillant peut supprimer une partie ou bien la totalité de paquets qu'il reçoit au lieu de les acheminer à leur destination. D'autre part, les nœuds d'un réseau MANET tendent à être égoïstes dû à leurs ressources limitées. Par conséquent, il est possible pour un nœud de ne pas coopérer dans le routage des paquets pour conserver ses ressources en termes de batterie et de bande passante.
La sécurité est un besoin fondamental dans les réseaux MANET qui peut être garantie grâce à des contre-mesures permettant d'empêcher les comportements malveillants de perturber la performance des services du réseau. Cependant, la sécurisation d'un réseau MANET ne peut pas être assurée par les mécanismes utilisés dans les réseaux sans fil traditionnels (ex., pares-feux), qui sont normalement conçu pour être utilisés dans des réseaux ayant une infrastructure centralisée. Dans le contexte de routage ad hoc, la sécurité a cinq objectifs essentiels : la *confidentialité* et l'*intégrité* de l'information de routage, l'*authentification* et la *non-répudiation* des nœuds et la *disponibilité* du réseau [3]. Ce dernier signifie que les services et les ressources du réseau doivent être disponibles même en présence des nœuds malveillants. Cette condition est assurée en protégeant le réseau contre les attaques de *Déni de Service* (DoS). Au niveau de routage ad hoc, ces attaques peuvent être classifiées en trois catégories :

- **Attaques par épuisement des ressources** : peuvent être réalisées par un attaquant même s'il ne fait pas partie du réseau en générant un taux excessif de paquets erronés pour consommer les ressources des nœuds du réseau.

- **Attaques par manipulation de trafic** : exploitent les vulnérabilités du protocole de routage pour attirer le trafic ou établir des liens malicieux entre les attaquants. Par conséquent, le trafic intercepté est susceptible d'être altéré ou bien supprimé selon les objectifs des attaquants.

- **Attaques par suppression des paquets** : dans cette classe d'attaques, quand un nœud malveillant reçoit des paquets qui sont supposés d'être acheminés, il supprime une partie ou bien la totalité de ces paquets pour éviter leur livraison normale à leur destination.

Dû à leur impact sévère sur la performance des réseaux MANET, les attaques par suppression des paquets ont attiré une attention considérable de plusieurs travaux de recherche. Parmi les solutions qui ont été proposées dans ces travaux, nous distinguons les mécanismes cryptographiques qui sont utilisés comme des stratégies préventives assurant l'authentification des nœuds et l'intégrité de l'information de routage [4],[5]. D'autres travaux de recherche ont montré que les systèmes de gestion de la confiance sont essentiels dans les réseaux MANET, surtout quand les

nœuds désirent établir un réseau avec un certain niveau de confiance entre eux sans aucune interaction précédente [6],[7].

En se basant sur des études précédentes [8], nous considérons que les solutions cryptographiques ne sont pas suffisantes pour prévenir tous les problèmes de sécurité causés par les nœuds malveillants. D'autre part, nous considérons que la performance des systèmes de confiance dans un réseau MANET est susceptible d'être dégradée dû à la topologie dynamique de ces réseaux. Par conséquent, notre objectif est de proposer une solution légère et totalement décentralisée pour surmonter les défis qui font face à la plupart des solutions utilisant la cryptographie et les systèmes de confiance. Nous explorons certaines techniques statistiques et probabilistes pour concevoir et implémenter un mécanisme de détection contre les attaques par suppression des paquets dans les réseaux MANET. Nous considérons que telle solution fait face à plusieurs défis techniques :

- **Décentralisation** : l'absence d'une administration centrale dans les réseaux MANET compliquent la détection des comportements malveillants. Comment peut-on assurer que toutes les phases de détection sont décentralisées?
- **Mobilité des nœuds** : la mobilité imprévisible des nœuds cause des changements fréquents dans la topologie du réseau, ce qui peut affecter la précision de détection des nœuds malveillants. Comment peut-on concevoir un mécanisme de détection qui est adapté à une telle topologie?
- **Contraintes des ressources** : les nœuds d'un réseau MANET tendent à se comporter d'une manière égoïste pour conserver leurs ressources en termes de batterie et bande passante. Quelles sont les caractéristiques d'un mécanisme de détection qui prend en considération les limitations des ressources?
- **Précision de détection** : le mécanisme de détection doit être basé sur un ensemble d'attributs permettant de réaliser une analyse comportementale des nœuds du réseau. Comment peut-on concevoir une méthode de modélisation des comportements des nœuds permettant de détecter ceux qui sont malveillants avec une haute précision?
- **Profil d'attaquant** : une nœud malicieux peut changer ses comportements en fonction du temps en supprimant des paquets parfois et les acheminant d'une manière normale d'autres fois pour tromper le mécanisme de détection utilisé. Comment peut-on exploiter l'évolution d'un nœud en fonction du temps pour reconnaître la nature de ses comportements?

Contributions

La première contribution de cette thèse consiste à proposer une nouvelle taxonomie d'attaques qui menaçent la disponibilité des services de routage ad hoc. Les scénarios d'attaques qui peuvent conduire à un Déni de Service sont classifiés

en trois catégories : les attaques par suppression des paquets, les attaques par épuisement des ressources et les attaques par manipulation de trafic. Pour chaque catégorie d'attaques, nous explorons les différentes techniques, les objectifs et l'impact sur la performance de routage.

La deuxième contribution consiste à proposer un mécanisme de détection décentralisé contre les attaques par suppression des paquets dans les réseaux MANETs. Ce mécanisme est basé sur une classification probabiliste des nœuds du réseau pour déterminer la nature de leurs comportements et décider s'ils sont fiables pour acheminer les paquets. Nous utilisons le mode promiscuité pour permettre à un nœud d'écouter les paquets passant par ses nœuds voisins quelle que soient leurs destinations [9]. Nous modélisons les comportements d'un nœud sous forme d'un vecteur ayant trois attributs représentant les taux de transmission de trois types des paquets : les paquets de données *Data*, les paquets de demande de route RREQ (Route Request) et les paquets de réponse de route RREP (Route Reply). En se basant sur les valeurs de ces attributs nous classifions un vecteur de comportements en évaluant sa probabilité de malveillance. Cette valeur de probabilité est obtenue par un calcul Bayésien basé sur deux modèles de classification : Bernoulli et Multinomial. Nous utilisons une métrique nommée TCR (Total Cost Ratio) qui permet d'évaluer la performance de ces modèles en comparant leurs taux d'erreur avec celui d'une ligne de base. Les résultats de simulation montrent que l'utilisation des modèles Bayésiens peut assurer la détection des comportements malveillants dans les réseaux MANET avec une haute précision. Nous constatons que l'utilisation du modèle de Bernoulli assure une détection complète des nœuds malveillants quand leur pourcentage dans le réseau est inférieur à 15%, alors que l'utilisation du modèle Multinomial est plus appropriée quand ce pourcentage excède une valeur de 27%. Nous déduisons que les deux modèles peuvent être utilisés d'une manière complémentaire pour assurer une détection complète des attaques par suppression des paquets dans les réseaux MANET.

La troisième contribution est une extension de la première approche, dans laquelle nous proposons de calculer la probabilité de malveillance d'un nœud surveillé d'une manière périodique (à chaque $t = \tau$) afin de faire la traçabilité de ses comportements durant une période T. Nous utilisons un modèle basé sur la *logique floue* pour associer une valeur linguistique à chaque valeur de probabilité obtenue durant une période τ. Nous définissons trois niveaux de malveillance représentant l'espace fini des états des comportements possibles d'un nœud : *légitime*, *suspect* et *malveillant*. Nous utilisons la séquence des transitions entre les différents états pour représenter l'évolution d'un nœud sous forme d'une matrice stochastique en utilisant les *chaînes de Markov*. Nous démontrons mathématiquement que chaque nœud du réseau admet un état stationnaire qui peut être prédit en appliquant le théorème d'*ergodicité* avec un nombre d'étapes inférieur à 10. Nous expliquons l'importance de l'utilisation de la classe de comportements suspects (i.e., non-détecté) pour surmonter les défis causés par les attaques par *suppression périodique des paquets*.

Les résultats de simulation montrent que notre solution est capable de détecter les différents types d'attaques par suppression des paquets avec une valeur moyenne de précision supérieure à 90% dans des différentes configurations du réseau en termes des : taille du réseaux, pourcentage des nœuds malveillants et durées des simulations.
Dans le reste de ce résumé, nous allons tout d'abord présenter les différentes applications des réseaux MANET, les concepts fondamentaux de routage dans ces réseaux et les défis de sécurité menaçant le routage ad hoc. Par la suite, nous développons un état de l'art analysant des solutions existantes pour la sécurisation de routage ad hoc contre les attaques DoS. Nous présentons les solutions que nous proposons pour lutter contre les attaques par suppression des paquets. Finalement, nous terminons ce résumé avec une conclusion dans laquelle nous discutons les défis qui peuvent faire face à nos solutions, et nous présentons nos perspectives de recherche à court terme et à long terme.

Réseaux MANETs : applications et challenges de sécurité

Avec l'évolution des exigences des utilisateurs, des nombreuses technologies de réseau ont été développées en se basant sur le concept de communication *ad hoc*. Parmi ces technologies, les réseaux mobiles ad hoc (MANET) ont été conçus pour assurer une communication fiable où il est inapproprié ou coûteux de déployer une infrastructure réseau. Ces réseaux ne s'appuient sur aucune infrastructure fixe, où les nœuds sont chargés d'agir d'une manière coopérative pour assurer les différents services du réseau.

Applications de réseaux MANETs

Les réseaux MANET ont vu le jour dans le domaine militaire. À cette époque, ces réseaux étaient largement employés comme des moyens de communication dans des environnements rigoureux. Leur déploiement facile, rapide et non-coûteux était bien adapté pour assurer la communication entre les unités militaires dans les champs de bataille où il est difficile de déployer une infrastructure réseau.
Dans la dernière décennie, la révolution de technologies sans fil et l'émergence des équipements mobiles ont impliqué l'utilisation de la communication ad hoc dans un large nombre des applications et assurer l'amélioration de la qualité de plusieurs services urbains avec des coûts réduits :

- **Réseaux véhiculaires (VANET)** : les réseaux VANET (Vehicular Ad hoc Network) sont des réseaux ad hoc utilisés pour assurer la communication au sein d'un groupe de véhicules en deux types [10]. Le premier type est la communication véhicule à véhicule, alors que le deuxième type est défini par une communication entre les véhicules et des équipements de route nommés RSU (Road Side Unit). Les réseaux VANETs sont largement utilisés pour implémenter des systèmes de sécurité routière, des services de confort et des solutions de stationnement intelligentes.

- **Réseaux de capteurs (WSN)** : comme leur nom l'indique, les WSN (Wireless Sensor Network) sont basés sur des capteurs sans fil qui sont capables de récolter et de transmettre des données au sein d'un système environnemental, domestique ou sanitaire [11]. Ces données sont traitées et analysées par des unités nommés *stations de base*, afin d'améliorer les conditions d'utilisation du système étudié.

- **Réseaux maillés (WMN)** : c'est une topologie réseau connue par WMN (Wireless Mesh Network) dont toutes les entités sont connectées en mode ad hoc [12]. Parmi leurs objectifs principaux, ces réseaux servent à reproduire l'architecture de l'Internet tout en l'optimisant pour une communication sans fil. Les réseaux Mesh font partie aujourd'hui de la vie quotidienne ; ils permettent de connecter des zones encore blanches et de déployer l'Internet dans des domaines et des situations où les réseaux actuels font défaut.

- **Réseaux personnels (WPAN)** : ou bien les réseaux domestiques sans fil sont connus par WPAN (Wireless Area Personal Network) permettent de connecter des équipements sans fil ayant une portée de transmission limitée à une dizaine de mètres [13]. La technologie SPAN (Smart Phone Ad hoc Network) est une parmi les applications de WPAN qui sert à créer une communication sans fil entre deux smartphone sans passer par un réseau d'opérateur. Les réseaux WPAN sont largement employés pour assurer une communication en champ proche NFC (Near Field Communication), qui incrustent de l'intelligence un peu partout dans des applications quotidiennes comme les cartes de paiement sans contact et les puces d'identification RFID (Radio Frequency Identification).

- **Informatique ubiquitaire** : le terme ubiquitaire fait référence à l'omniprésence de l'accès à l'information, c'est-à-dire, un accès n'importe où et n'importe quand, grâce à une gamme de petits appareils informatiques [14]. Parmi les applications ubiquitaires nous trouvons les maisons intelligentes connues par *Smart House* qui permettent par exemple d'ajuster un système de chauffage ou de climatisation, ou encore télécommander un décodeur télévision depuis une application mobile.

- **Applications médicales** : dans la dernière décennie, plusieurs systèmes de surveillance médicale et sanitaire ont vu le jour afin d'améliorer le mode de vie de l'homme [15]. Ces systèmes sont basés sur des capteurs qui sont généralement déployés et interconnectés sur, autour ou bien dans le corps humain pour surveiller certaines conditions sanitaires afin d'assurer la prévention contre certaines maladies. D'autre part, l'émergence des équipements mobiles a favorisé le développement des applications Web et mobile permettant de surveiller les conditions sanitaires d'un utilisateur depuis un smartphone ou bien une montre intelligente, afin de satisfaire certains facteurs de confort comme le régime alimentaire et l'analyse du sommeil.

Protocoles de routage ad hoc

Le routage est une fonction primordiale dans les réseaux MANET, puisqu'il constitue la base pour l'échange des données entre les nœuds mobiles. Les caractéristiques intrinsèques de ces réseaux ont imposé l'utilisation des nouveaux protocoles de routage qui assurent des approches distribuées d'établissement des routes entre les nœuds, tout en considérant le changement de topologie et les limitations de ressources.
En fonction du mode de fonctionnement de l'établissement et la maintenance de route, les protocoles de routage ad hoc peuvent être classifiés en trois catégories :

- **Réactif** : ces protocoles sont à la demande, où les nœuds échangent les informations de routage seulement quand il y a un besoin de découverte de route.
- **Proactif** : les nœuds échangent entre eux des informations périodiques sur la topologie du réseau pour que toutes les routes soient disponibles à tout moment.
- **Hybride** : ce type combine les deux approches précédentes ; il adopte une méthode proactive pour établir les chemins à l'avance dans un voisinage à un nombre limite de sauts, et utilise une méthode réactive au delà de cette limite.

Dans ce qui suit, nous allons décrire le protocole de routage AODV (Ad hoc On-demand Distance Vector), que nous allons utiliser dans les approches de détection proposées dans cette thèse.

Le protocole AODV

AODV est un protocole de routage basé sur un algorithme de recherche de route réactif qui a été proposé dans [16]. Quatre principaux types de messages de contrôle de routage sont définis dans ce protocole : RREQ , RREP, RERR (Route ERRor) et Hello. Ces messages sont utilisés respectivement pour demander une route, répondre à une demande de route, signaler une erreur ou une rupture de lien et maintenir les routes établies.
Selon ce protocole, lorsqu'un nœud source a besoin d'avoir une route vers une certaine destination et qu'aucune route n'est disponible, il diffuse un message RREQ à tous les nœuds voisins qui sont dans sa portée de transmission. Un nœud intermédiaire recevant ce message vérifie s'il a une route vers la destination demandée, et envoie un message RREP vers la source, le cas échéant. Sinon, il enregistre une route inverse vers la source, incrémente son nombre de sauts du message RREQ et le rediffuse vers ses voisins. Cette procédure est établie jusqu'à trouver une route valide vers la destination. Si c'est le cas, le nœud destination envoie un message de réponse de route RREP vers le nœud source en passant par les nœuds constituant la route inverse qu'il l'a extrait du paquet RREQ reçu. Lorsque le message RREP atteint la source, un chemin bidirectionnel est établi entre la source et la destination et la transmission de paquets de données peut débuter.
La maintenance des routes découvertes est assurée par une transmission périodique

de messages Hello. Si au bout d'une certaine période, aucun message Hello n'est reçu d'un nœud voisin, le lien correspondant est considéré comme défaillant. Dans ce cas, un message d'erreur RERR se propage vers la source et tous les nœuds intermédiaires afin de marquer la route comme invalide.
AODV utilise le principe du numéro de séquence pour garantir la consistance des informations de routage pour que les routes empruntées par les nœuds soient valables et fraîches (à jour). D'autre part, ce protocole a l'avantage de réduire le nombre de messages de contrôle échangés étant donné que les routes sont créées à la demande. Cependant, la phase d'établissement de route peut engendrer des délais importants avant la transmission des données. En plus, AODV est vulnérable aux attaquants qui peuvent exploiter la politique de "diffusion" utilisée dans la phase de découverte de route pour inonder le réseau par des messages erronés afin d'épuiser ses ressources et perturber ses services. Ces comportements malveillants et d'autres vulnérabilités de sécurité de routage ad hoc sont abordés dans la section suivante.

Taxonomie d'attaques DoS au niveau du routage ad hoc

Assurer un routage sécurisé dans les réseaux MANET est une problématique de recherche qui attire une attention particulière de plusieurs communautés de recherche. La majorité des protocoles de routage existants assument que les nœuds du réseau sont dignes de confiance et coopératifs. Par conséquent, les services de routage sont menacés par plusieurs attaques de sécurité, allant de l'écoute passive jusqu'aux attaques de Déni de Service (DoS). Ce dernier comporte l'ensemble de comportements malveillants qui peuvent perturber la disponibilité des services de routage ad hoc.

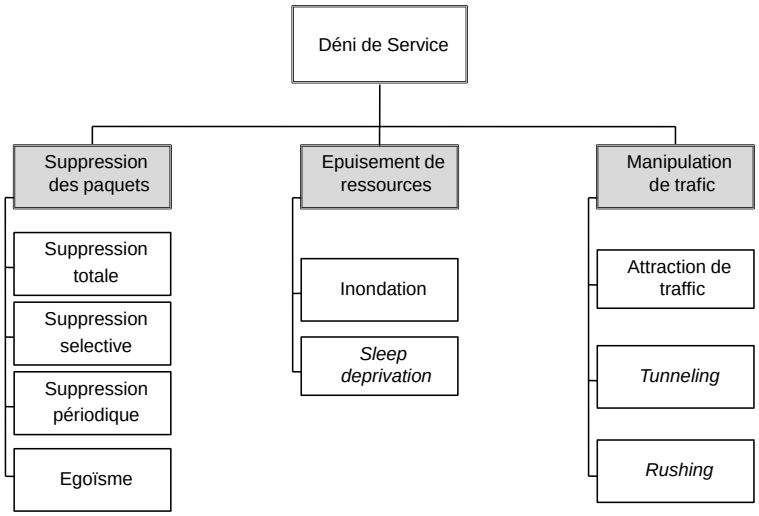


Figure 1: Taxonomie d'attaques DoS au niveau de routage ad hoc.

Dans notre étude bibliographique, nous avons fait une recherche extensive sur les

différents scénarios d'attaques qui peuvent engendrer un Déni de Service. Dans la figure 1, nous proposons une nouvelle taxonomie d'attaques DoS, qui définit trois catégories d'attaques qui menaçent la disponibilité de routage ad hoc : les attaques par suppression des paquets, les attaques par épuisement des ressources et les attaques par manipulation de trafic. Nous effectuons pour chaque catégorie d'attaques une simulation avec NS2 (Network Simulator 2) afin de montrer le taux de perte des paquets causée par ces attaques. Les figures 2, 3 et 4 montrent l'impact de ces attaques sur la performance d'un réseau MANET en terme de taux de perte des paquets selon les résultats que nous avons obtenu à travers des simulations.

- **Attaques par suppression des paquets** : la suppression des paquets est une violation sévère de la propriété de coopération entre les nœuds pour assurer les services de routage. La manière de suppression des paquets dépend de l'objectif de l'attaquant, qui peut supprimer la totalité de paquets afin d'interrompre les connexions entre les nœuds communiquant, ou bien une partie de ces paquets pour compliquer sa détection. La suppression des paquets peut être aussi effectuée par des nœuds ayant des objectifs égoïstes qui consistent à conserver leurs ressources aussi longtemps que possible.

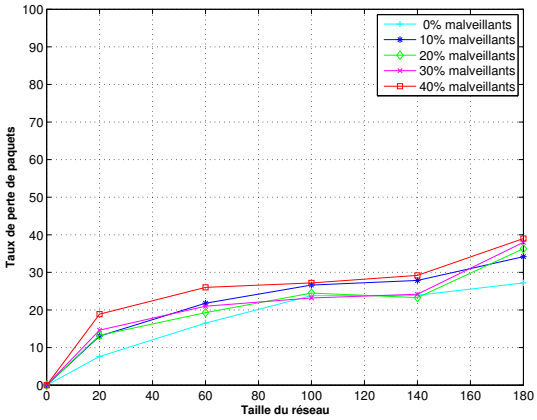


Figure 2: Taux de perte causée des paquets par les attaques par suppression.

- **Attaques par épuisement de ressources** : ces attaques sont effectuées par des entités internes ou externes qui peuvent envoyer un taux excessif des paquets inutiles aux nœuds du réseau pour épuiser leurs ressources. Parmi ces attaques nous distinguons l'attaque par inondation RREQ qui menacent généralement les protocoles de routage réactifs (ex., AODV). Cette attaque consiste à bloquer le réseau entier en générant des demandes des routes inutiles avec une fréquence excédant celle qui est définie dans les spécifications du protocole. Par conséquent, le réseau entier est susceptible d'être inondé dû à l'obligation du protocole qui consiste à rediffuser les messages RREQ reçus. Une entité malveillante peut aussi faire des interactions inutiles avec un ou plusieurs nœuds du réseau afin de les occuper d'une manière continue

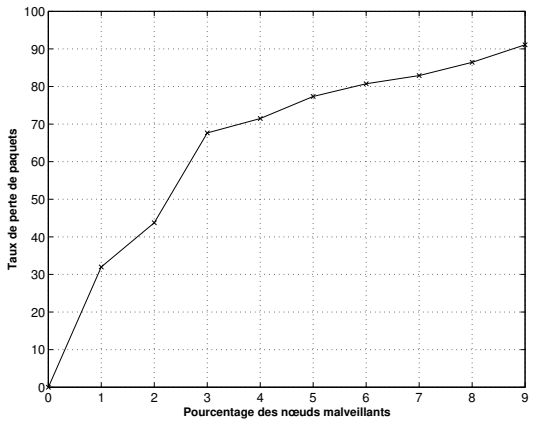


Figure 3: Taux de perte des paquets causée par les attaques par épuisement des ressources.

et causer ce qu'on appelle *sleep deprivation*.

- **Attaques par manipulation de trafic** : regroupent les comportements malveillants qui ont pour objectif de perturber l'exécution normale d'un protocole de routage, en déroutant les paquets selon les objectifs de l'attaquant. Par exemple, un attaquant peut attirer des nœuds dans sa portée de transmission en leur faisant une sorte d'illusion qu'il a une route vers les destinations qu'ils demandent. Un autre scénario possible consiste à créer un tunnel malveillant entre deux nœuds distants du réseau pour tromper d'autres nœuds et les forcer à emprunter une route malveillante pour envoyer leurs paquets. Une autre attaque ayant le même objectif de l'attaque *tunneling* connue par l'attaque *rushing*. Cette attaque consiste à tromper les nœuds du réseaux en livrant leurs paquets aux destinations désirées aussi vite que possible.

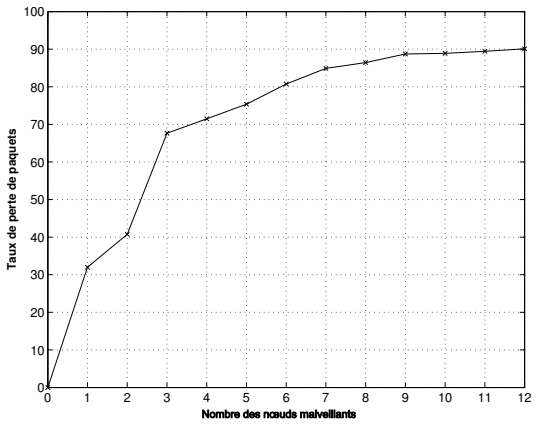


Figure 4: Taux de perte des paquets causée par les attaques par manipulation de trafic.

État de l'art de sécurité du routage ad hoc

La sécurité est un sujet important à traiter, surtout pour les applications de MANET qui sont souvent sensibles à la sécurité comme les applications militaires. Dans cette section, nous présentons pour chaque catégorie d'attaques DoS de la taxonomie déjà présentée certains mécanismes de sécurité qui ont été proposés pour prévenir, détecter ou isoler les nœuds malveillants dans les réseaux MANETs.

Mécanismes contre les attaques DoS par suppression des paquets

Les mécanismes de sécurité proposés pour lutter contre ces comportements malveillants peuvent être classifiés en trois grandes catégories: les mécanismes cryptographiques, les systèmes de gestion de la confiance et les mécanismes de détection. Dans ce qui suit, nous citons pour chaque catégorie certain nombre de travaux de recherche les plus connus dans la littérature.

- **Mécanismes cryptographiques** : la plupart de ces solutions consistent à proposer des méthodes permettant d'assurer l'authentification des nœuds et l'intégrité des informations de routage afin de prévenir l'occurrence des attaques par suppression des paquets. Parmi ces solutions, des protocoles de sécurité ont été proposés [17], [18] pour assurer l'intégrité des messages de routage en utilisant le chiffrement à clé publique. L'objectif de ces protocoles est de garantir la meilleure exécution de la phase de découverte de route par les nœuds du réseau. Dans [19], les auteurs ont proposé une méthode qui s'appelle le chiffrement adaptatif des messages de routage, qui consiste à assigner un niveau de chiffrement pour un nœud selon son niveau de confiance. Une autre solution proposée dans [4] consiste à assurer une authentification saut-par-saut des messages de routage pour assurer leur intégrité. D'autres solutions comme [20] et [21] ont utilisé le chiffrement symétrique pour sécuriser la phase de découverte de route dans le protocole AODV.

- **Systèmes de confiance** : la plupart des solutions basées sur les systèmes de confiance ont pour objectif de forcer les nœuds ayant des comportements malveillants à participer dans le routage selon les spécifications du protocole. Les systèmes de confiance ont été utilisés dans plusieurs travaux de recherche comme [22], [23] et [24], pour permettre à un nœud de vérifier si les paquets envoyés vers un autre nœud voisin sont bien transmis ou supprimés. Selon les comportements observé, un niveau de confiance est assigné au nœud surveillé. Le niveau de confiance peut être obtenu soit via l'observation directe d'un nœud ou bien par une combinaison entre cette observation et les recommandations des autres nœuds. D'une manière similaire, la solution proposée dans [7] utilise un système de réputation qui prend en considération la qualité des liens comme un paramètre de qualité de service pour choisir les routes. Les travaux proposés dans [25] et [26] consistent à améliorer la performance du protocole AODV en utilisant un système de confiance.

- **Mécanismes de détection** : ces mécanismes ont pour objectif de détecter les nœuds malveillants en se basant sur une analyse comportementale. Les auteurs de [27] ont proposé un mécanisme de détection des nœuds malveillants en utilisant une machine à états finis permettant de modéliser leurs comportements. Leur solution est basée sur l'utilisation des matrices de transitions pour faire une traçabilité des nœuds pour détecter ceux qui sont malveillants. Dans [28], une méthode de classification supervisée pour détecter les attaques par suppression a été proposée. Un modèle de distribution normale des paquets RREQ et RREP a été défini et utilisé pour détecter les nœuds comportant des déviations de ce modèle. Dans [29], les auteurs ont proposé un modèle de prédiction de confiance des nœuds qui se base sur l'historique de leurs comportements. Ces derniers sont modélisés suite à une analyse des paquets des données et de routage échangés par les nœuds. Un modèle de routage fiable de paquets de données a été présenté dans [30]. L'idée principale de ce modèle consiste à évaluer la probabilité qu'un nœud soit un attaquant en analysant les paquets RTS (Request-To-Send) et CTS (Clear-To-Send) qu'il échange.

Mécanismes contre les attaques DoS par épuisement de ressources

Malgré leur impact sévère sur les services de routage, il n'y a pas suffisamment des travaux de recherche qui ont traité les attaques DoS par épuisement des ressources. La plupart de ces travaux proposent des solutions pour protéger les protocole de routage réactifs contre les attaques par inondation de RREQ. Plusieurs améliorations des protocole existants ont été proposées pour lutter contre les nœuds qui génèrent un taux excessif de messages RREQ. Dans certaines versions des protocoles réactifs, plusieurs spécifications ont été implémentées pour mitiger les effets de ces attaques, comme la définition d'un taux limite de demandes de route dans le cas du protocole AODV [16].
Plusieurs solutions ont été proposées pour surveiller les nœuds du réseau, évaluer le taux de leurs demandes de route et vérifier la légitimité de ces demandes [31], [32], [33], [34]. D'autres solutions ont proposé des mesures réactives permettant d'isoler les nœuds réalisant ce type d'attaques [35], [36]. Finalement, des mécanismes de prévention ont été proposées afin d'éviter l'occurrence des attaques par inondation des paquets RREQ, en ajoutant des spécifications qui améliorent la protection des protocoles de routage existants [37], [38].

Mécanismes contre les attaques DoS par manipulation de trafic

Les attaques par manipulation de trafic ont été bien traitées dans plusieurs travaux de recherche qui ont proposé des mécanismes permettant de détecter, localiser et mitiger l'impact de ces attaques sur les réseaux MANETs. Ces mécanismes peuvent être classifiés comme suivant :

- **Analyse temporelle des paquets** : l'objectif de cette classe de solutions

est de permettre à un nœud de vérifier si les paquets qu'il les envoie passent par des nœuds appartenant à sa propre portée de transmission. La plupart de ces solutions ajoutent une information temporelle à l'envoie et à la réception des paquets pour vérifier s'ils sont acheminés via des routes truquées [39], [40]. D'une manière similaire, d'autres solutions ont essayé de définir une relation entre le temps de transmission de paquets et le nombre de sauts afin de détecter les tunnels créés par les nœuds malveillants [41]. D'autre part, le temps d'aller-retour des paquets qui est bien connu par RTT (Round Trip Time) a été exploité dans plusieurs travaux de recherche pour détecter les liens Wormhole dans les réseaux MANETs [42].

- **Mécanismes de localisation** : ces mécanismes utilisent des informations spatio-temporelles des paquets échangés entre les nœuds pour identifier et localiser ceux qui réalisent des attaques Wormhole. Dans [43], les auteurs ont proposé un mécanisme de vérification sécurisée des temps des rencontres entre les nœuds pour les distances qui les séparent, et ensuite détecter ceux qui établissent des connexions malveillantes dans le réseau. Une autre solution pour lutter contre ces attaques a été proposée dans [44], en utilisant des équipements spécialisés nommés antennes directionnels. L'idée de cette solution consiste à vérifier la consistance de direction des ondes radios échangées entre les nœuds.

- **Détection basée sur la connectivité** : les liens malicieux créés par les attaques Wormhole ont été traités dans plusieurs approches utilisant des informations sur la connectivité du réseau et analysé d'un point de vue géométrique. Dans [45], les auteurs ont présenté comment ces attaques peuvent être détectées en utilisant les graphes géométriques aléatoires. Ils ont montré que le changement de topologie causé par une attaque Wormhole peut être détecté avec ce modèle. D'autres approches de détection de ces attaques ont été proposé dans [46][47].

Approche de détection des nœuds malveillants

L'objectif de cette approche est d'assurer une détection des nœuds malveillants dans un réseau MANET, et précisément ceux qui ne participent pas dans le routage en supprimant les paquets au lieu de les acheminer à leurs destinations. Notre idée consiste à effectuer une analyse comportementale des nœuds en se basant sur les taux de paquets qu'ils ont transmis durant une période de monitorage. La structure générale de notre système de détection est illustrée dans la figure 5.

La détection est effectuée d'une manière décentralisée, où chaque nœud du réseau est chargé de surveiller ses nœuds voisins afin de sélectionner ceux qui peuvent assurer un routage fiable des paquets. Nous présentons dans ce qui suit les étapes nécessaires détecter les attaques par suppression des paquets avec une haute précision :

- **Monitorage périodique** : durant une période définie, chaque nœud collecte

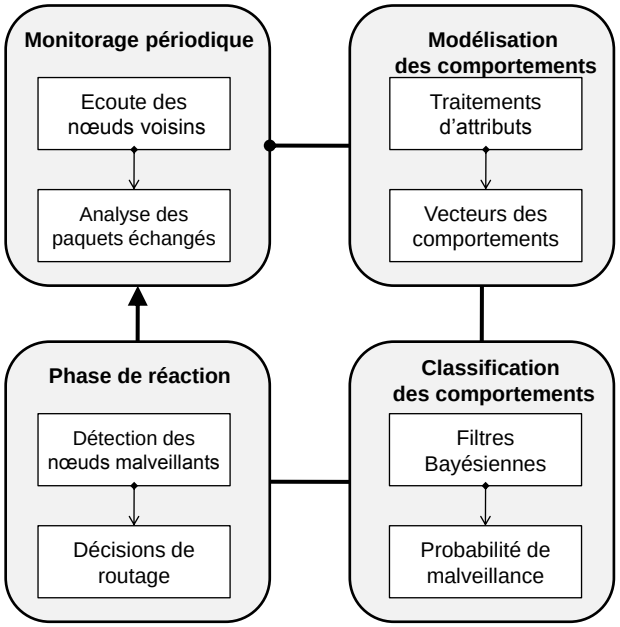


Figure 5: Phases du système de détection.

des informations sur les paquets échangés par un voisin surveillé, en utilisant le mode promiscuité qui permet d'écouter le trafic passant par les nœuds qui sont dans sa portée de transmission.

- **Modélisation des comportements** : les informations collectées sont utilisées pour créer un modèle de comportements du nœud surveillé. Pour sélectionner les attributs nécessaires pour la modélisation de comportement, nous évaluons la performance de nœud en ce qui concerne trois attributs qui sont associés à trois types de paquets :

  - *DATA* : la fiabilité de nœud à acheminer les données entre les entités en communication, en évaluant le taux des paquets de données qui les a correctement transmis.
  - *RREQ* : le participation de nœud à la phase de découverte des routes en évaluant le taux de messages RREQ qui sont correctement rediffusés par le nœud.
  - *RREP* : l'aptitude de nœud à assurer l'établissement des routes demandées en évaluant le taux de messages RREP qui les a transmis.

  En se basant sur ces trois attributs, nous représentons les comportements d'un nœud sous forme d'un vecteur composé de trois éléments, où chacun sert à évaluer le taux de transmission d'un type de paquet. Dans notre cas, nous utilisons deux types de vecteurs. Le premier type représente le modèle de comportements avec des valeurs booléennes, où chaque élément indique

l'occurrence ou l'absence d'une suppression d'un type de paquet. Dans le deuxième type, les éléments des vecteurs sont des pourcentages qui évaluent les taux de transmissions pour chaque type de paquets.

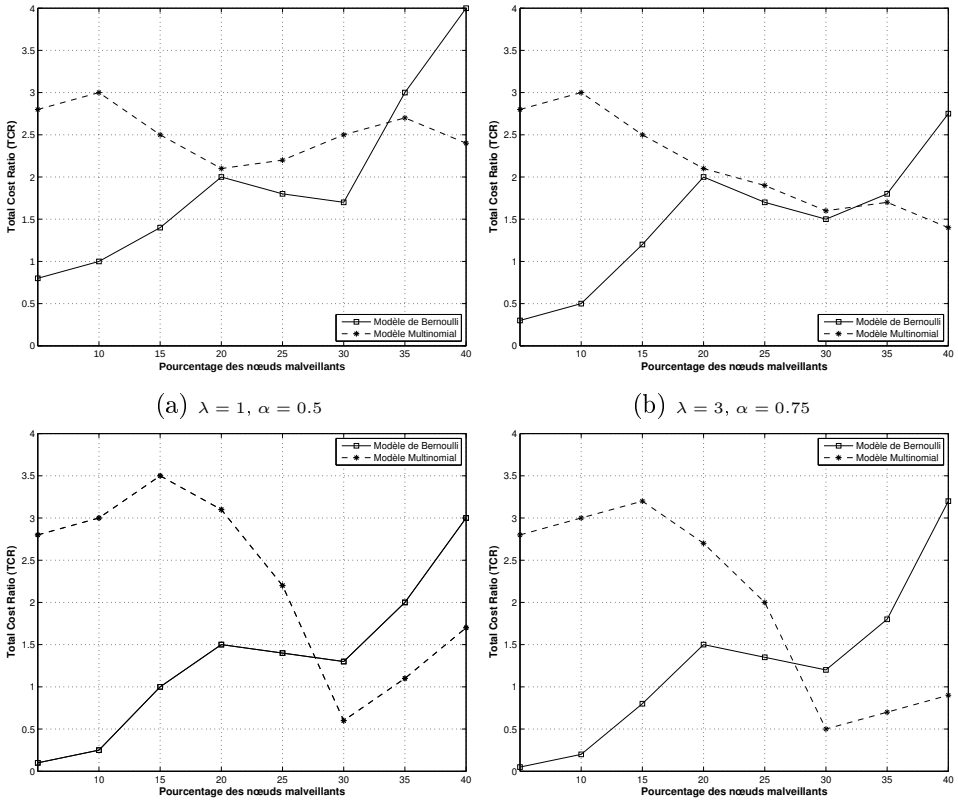
- **Classification des comportements** : le vecteur de comportements d'un nœud obtenu dans la phase précédente est utilisé pour calculer sa probabilité de malveillance selon deux modèles de classification Bayésienne : Bernoulli et Multinomial. Dans le modèle de Bernoulli, le calcul de probabilité est basé sur des vecteurs booléens, alors que le modèle Multinomial permet de calculer la probabilité de malveillance d'un nœud en utilisant le deuxième type de vecteurs.

- **Phase de réaction** : la valeur de probabilité obtenue est comparée avec le seuil de classification du modèle nommé α. Si la probabilité de malveillance est supérieure à α le nœud est classifié comme malveillant, et comme légitime ailleurs.

Évaluation de l'approche proposée

Pour montrer la précision de détection assurée par notre approche nous introduisons un paramètre qui permet d'évaluer les modèles de classification utilisés. Ce paramètre est nommé TCR (Total Cost Ratio) qui sert à comparer l'erreur causée par un filtre de classification avec celui d'une ligne de base, c'est-à-dire, sans utiliser un filtre. Un modèle de classification est considéré important si son TCR est supérieur à 1, et il n'a aucune valeur ailleurs.
Nous utilisons un deuxième paramètre nommé λ pour ajuster la valeur du seuil de détection α suivant la formule $\alpha = \lambda/\lambda + 1$. λ est utilisé pour donner un rapport des coûts entre les faux positifs et les faux négatifs. Un faux positif dans notre cas signifie une *erreur de classification d'un nœud légitime comme malveillant*, alors qu'un faux négatif signifie une *erreur de classification d'un nœud malveillant comme légitime*. Dans notre cas, nous utilisons ce rapport pour donner plus d'importance aux faux positifs. Par exemple, si λ est égal à 5 alors les faux positifs sont 5 fois plus coûteux que les faux négatifs. Avec une valeur de λ égale à 1, les deux erreurs ont le même coût, dans ce cas la valeur par défaut du seuil de détection α qui est égal à 0.5.
Pour évaluer la performance de notre classification, nous calculons la valeur du TCR selon quatre configurations des seuils de détection α en fonction de quatre valeurs de λ comme le montre la figure 6. L'analyse de performance est validée via des simulations effectuées avec le simulateur NS2.

Selon la figure 6a, quand le seuil de détection α est égal à 1, les deux modèles montrent une haute performance quel que soit le pourcentage des nœuds malveillant dans le réseau. Cependant, le modèle de Bernoulli n'a aucune importance pour détecter les faibles pourcentages des nœuds malveillants ($< 10\%$). En plus, dans les autres configurations de α, le modèle de Bernoulli persiste incapable de détecter les



(a) $\lambda = 1$, $\alpha = 0.5$ (b) $\lambda = 3$, $\alpha = 0.75$

(c) $\lambda = 9$, $\alpha = 0.9$ (d) $\lambda = 19$, $\alpha = 0.95$

Figure 6: Performance des modèles de classification en fonction du seuil de α

pourcentages faibles de nœuds malveillants avec une valeur de TCR inférieure à 1. Le modèle Multinomial montre une valeur de TCR qui est toujours élevée dans toutes les configurations du seuil α, et assure une détection des nœuds malveillants quel que soit leur pourcentage dans le réseau. Cependant, l'allure de la courbe TCR de ce modèle montre une diminution notable dans les deux dernières configurations, surtout quand le pourcentage des nœuds malveillants dans le réseau est notable ($> 27\%$).

Finalement, nous trouvons que les deux modèles de classification proposés dans notre approche montrent une bonne performance et assurent une détection totale des comportements malveillants s'ils sont utilisés en conjonction. Autrement dit, si le modèle de Bernoulli est utilisé dans le cas de pourcentage élevé, et le modèle Multinomial dans le cas de faible pourcentage des nœuds malveillants. Dans la prochaine section, nous présentons une deuxième approche qui permet de suivre l'évolution d'un nœud pour prédire la nature de ses comportements et l'éviter s'il est malveillant. Nous utilisons encore la classification Bayésienne et nous préférons de nous orienter vers le modèle de Bernoulli, en essayant d'augmenter la précision de détection dans le cas des faibles pourcentages des nœuds malveillants dans le réseau.

Analyse stochastique pour la prédiction d'états des nœuds

Dans notre deuxième contribution, nous cherchons à trouver une méthode de prédiction d'états des nœuds en se basant sur la traçabilité de leurs comportements. L'idée consiste à utiliser une analyse stochastique de l'évolution des nœuds durant une période définie. Cette solution assure une détection des différents types d'attaques par suppression avec une haute précision, surtout celles qui changent leurs comportements en fonction du temps. La figure 7 montre le modèle d'attaque effectuant une suppression périodique et aléatoire de paquets en comparaison avec un modèle normal. Nous remarquons qu'il est important de faire une longue période de surveillance pour pouvoir détecter tel type de suppression, surtout qu'elle se comporte parfois d'une manière légitime. Pour adresser cette problématique, nous

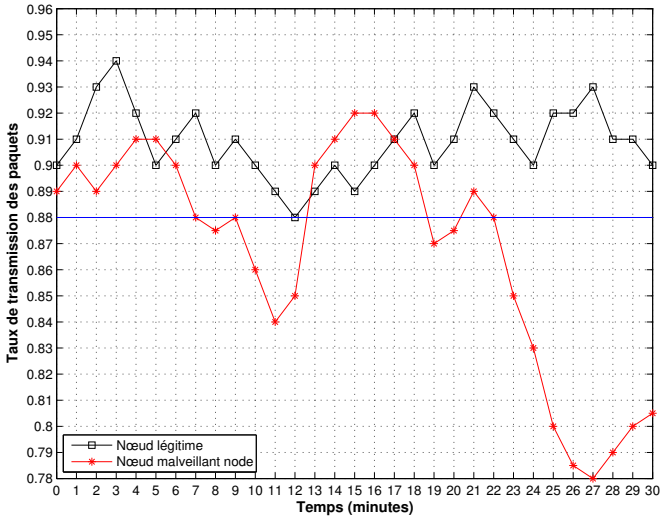


Figure 7: Illustration de la suppression périodique des paquets

introduisons le concept de comportements suspects ou encore incertains, qui seront clarifiés par la suite.
La figure 8 montre les composants de notre système de prédiction où quelques-uns sont similaires à ceux qui sont utilisés dans notre première approche.

- **Monitorage des nœuds** : ce module consiste à assurer un monitorage à long terme effectué par un nœud surveillant d'un autre nœud voisin dans sa portée de transmission. La durée de la phase de monitorage doit être suffisamment longue afin de constituer une base de connaissance suffisante sur les comportements des nœuds.

- **Sélection d'attributs** : cette étape consiste à collecter des informations sur les taux des paquets transmis pour les types Data, RREQ et RREP qui s'effectue exactement selon la méthode proposée dans la première approche.

- **Modélisation des comportements** : après avoir collecté les valeurs des attributs sélectionnés, les comportements sont modélisés en utilisant la même

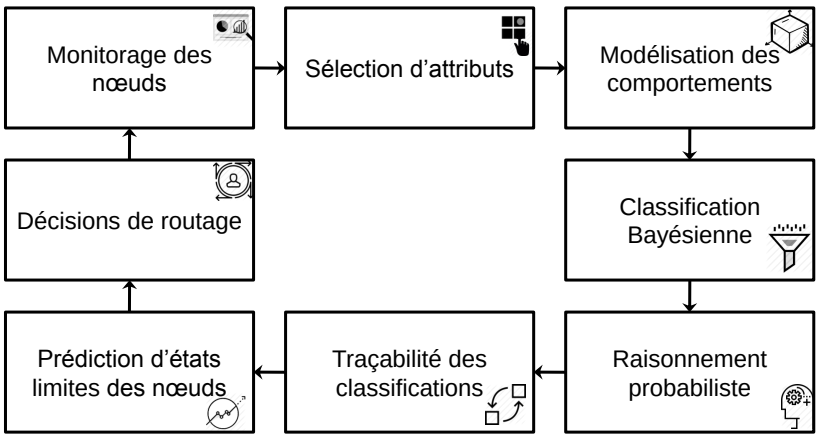


Figure 8: Composants de l'approche de prédiction de comportements.

méthode que nous avons décrit dans le système proposé dans la figure 5. Dans cette solution, nous utilisons les vecteurs Booléens pour calculer la probabilité de malveillance d'un modèle de comportements.

- **Classification Bayésienne** : la probabilité de malveillance d'un vecteur de comportements donné est calculée en utilisant le modèle de Bernoulli. Cependant, au lieu de faire référence au seuil de classification du modèle α pour vérifier si le nœud est malveillant, nous proposons d'assigner un niveau de malveillance selon deux valeurs seuils définissant une intervalle d'incertitude : $s = [\alpha_{min}, \alpha_{max}]$.

- **Logique floue** : nous proposons la définition de trois niveaux de malveillance associés aux valeurs de probabilité : légitime, suspect et malveillant. Ces comportements définissent un espace fini d'états possibles d'un nœud, et ils sont associés respectivement aux nœuds ayant une valeur de probabilité de malveillance inférieure à α_{min}, $\in [\alpha_{min} - \alpha_{max}]$ et supérieure à α_{max}.

- **Traçabilité des comportements** : l'affectation d'un niveau de malveillance à un nœud est réalisée à chaque intervalle de temps τ. Ensuite, la séquence d'états de comportements de ce nœud durant une période T est modélisé avec une chaîne de Markov, où les transitions entre ces états sont représentées sous forme d'une matrice stochastique ayant une taille 3×3. Dans cette matrice, un élément qui se trouve à la i^{me} ligne et la j^{me} colonne représente la probabilité que la chaîne se déplace de l'état i vers l'état j.

- **Prédiction d'états limites** : en appliquant le théorème d'ergodicité de la chaîne de Markov, nous prouvons que chaque nœud du réseau admet une distribution limite de se probabilité de transition qui définit son état stationnaire. Cet état est obtenu en calculant la puissance de la matrice jusqu'à l'obtention de trois lignes identiques, ce qui indique que l'état stationnaire est atteint.

- **Décisions de routage** : le vecteur de distribution limite d'un nœud est composé de trois valeurs qui sont associées respectivement aux probabilités d'un nœud d'être légitime, suspect et malveillant (P_l, P_s, P_m). Puisque la décision est basée sur ces trois valeurs, nous préférons d'utiliser la valeur maximale pour déterminer l'état final d'un nœud (voir figure 9). Dans ce cas, un nœud est considéré comme légitime si la valeur maximale est P_l (état S_3), et malveillant si la valeur maximale est P_m (état S_4). Si la valeur maximale est celle de P_s, une deuxième comparaison (état S_2) est faite entre P_l et P_m pour décider si cette incertitude dû à une suppression légitime ou malveillante des paquets. Finalement, si nous trouvons que les trois valeurs de probabilité sont égales ou bien la valeur de P_s est égale à 1, nous considérons que notre mécanisme est incapable d'identifier la nature des comportements observés.

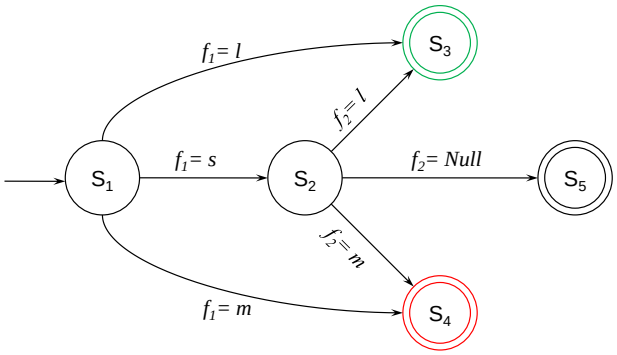


Figure 9: Le processus de décisions de routage.

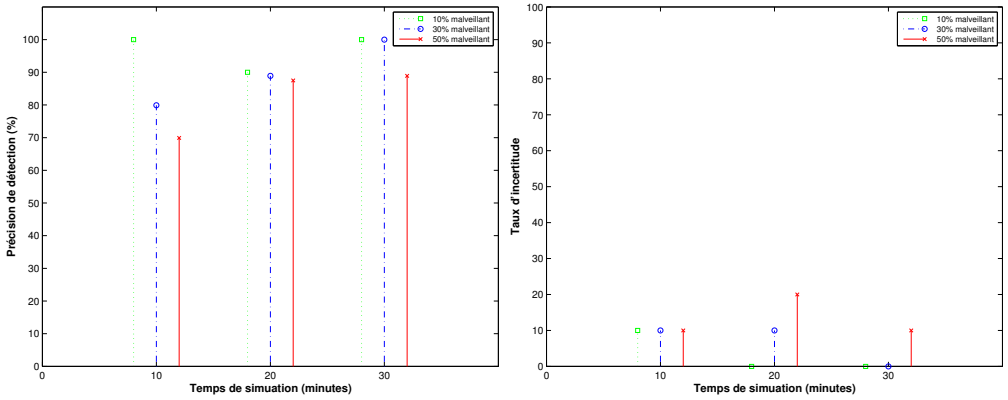
Évaluation de l'approche proposée

Après avoir décrit les différentes phases de notre mécanisme de détection, nous évaluons par la suite la performance de notre solution en termes de précision de détection. Le tableau 1 montre les différentes configurations des simulations que nous avons réalisé en utilisant le simulateur NS2.
Dans ce qui suit, nous analysons les résultats que nous avons obtenu pour évaluer la précision et l'incertitude de notre approche en fonction des différentes configurations.

Dans le cas où la taille du réseau est égale à 10 nœuds, la figure 10 montre un taux de précision qui atteint un pourcentage de 100% dans le cas d'une faible proportion des nœuds malveillants dans le réseau. Les valeurs de précision montrent une relation directement proportionnelle avec le temps de simulation et une relation inversement proportionnelle avec le pourcentage des nœuds malveillants. D'autre part, nous notons des valeurs élevées au niveau des taux d'incertitude, qui sont égales parfois à 20% surtout dans le cas d'une haute proportion des nœuds malveillants.

Table 1: Paramètres des simulations.

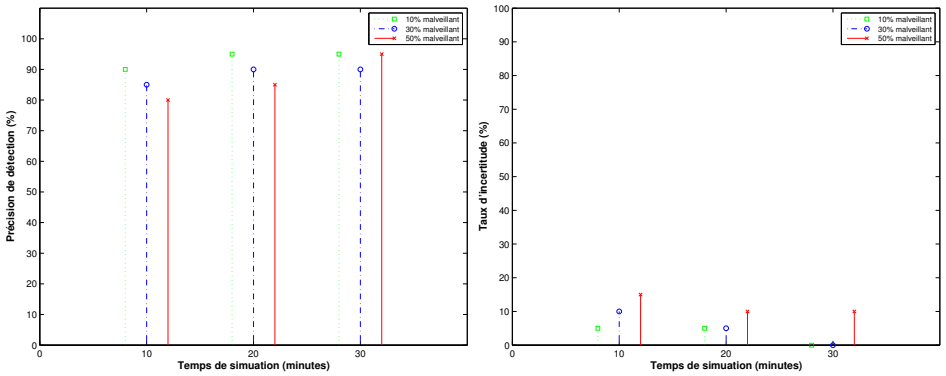
Paramètre	**Valeur**
Zone de couverture	1000m × 1000m
Nombre des nœuds	10, 20 et 50 nœuds
Portée de transmission	250m
Durée de simulation	10, 20 et 30 minutes
Laps de temps τ	1 minute
Modèle de mobilité	Random Waypoint
Antenne	Omnidirectionnel
Vitesse	$[2-8]$ m/s
Protocole de routage	AODV
Type de trafic	UDP/ CBR
% des nœuds malveillants	10, 30 et 50%
Intervalle d'incertitude	0.2
Couche physique	IEEE 802.11p



(a) Taux de précision vs temps de simulations (b) Taux d'incertitude vs temps de simulations

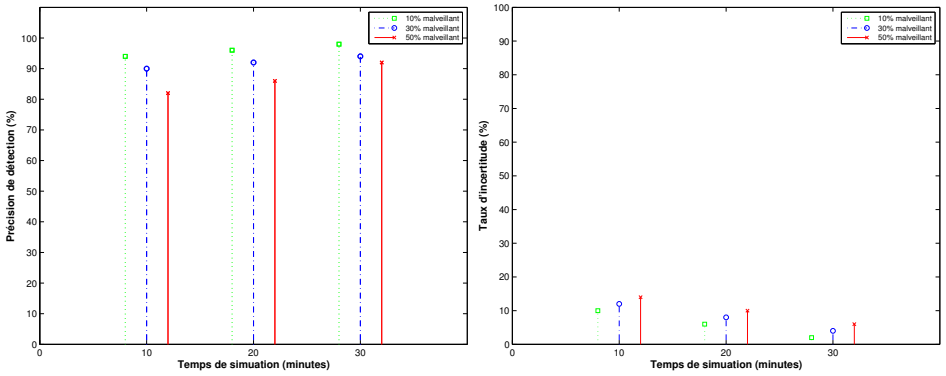
Figure 10: Taille du réseau = 10 nœuds

En augmentant la taille du réseau à 20 nœuds (figure 11), nous remarquons que les valeurs de précision sont inférieures à celles que nous avons obtenu dans le cas de 10 nœuds. Cependant, nous constatons une augmentation plus importante de la précision de détection en fonction de temps de simulation. Par exemple, dans le cas où le pourcentage des nœuds malveillants est égale à 50%, la précision de détection augmente 15% en étendant le temps de simulation de 10 à 30 minutes détection. En plus, nous remarquons une diminution notable au niveau de taux d'incertitude en comparaison avec le cas où la taille du réseau est égale à 10 nœuds. Finalement, quand nous augmentons la taille du réseau à 50 nœuds (figure 12), nous remarquons que les taux de précision de détection sont toujours supérieurs à 80% avec des valeurs qui sont inférieures à celles que nous avons obtenu dans les deux



(a) Taux de précision vs temps de simulations (b) Taux d'incertitude vs temps de simulations

Figure 11: Taille du réseau = 20 nœuds



(a) Taux de précision vs temps de simulations (b) Taux d'incertitude vs temps de simulations

Figure 12: Taille du réseau = 50 nœuds

cas précédents. Cependant, nous remarquons la variation de temps de simulation a plus d'impact sur les valeurs des taux de précision. Par exemple, dans le cas où le pourcentage des nœuds malveillants est 50%, la précision augmente de 80% à 95% lorsque le temps de simulation est augmenté de 10 à 30 minutes.

Conclusion et perspectives

Dans cette thèse, notre travail de recherche a été orienté pour adresser le problème des attaques DoS par suppression des paquets dans les réseaux MANET.
En premier, nous avons présenté une étude des attaques qui menacent la sécurité de routage ad hoc, et précisément celles qui donnent lieu à un déni de service. Nous avons proposé une nouvelle classification de ces attaques en trois grandes catégories : attaques par suppression des paquets, attaques par épuisement des ressources et attaques par manipulation de trafic. Nous avons étudié leurs techniques, leur objectifs et leurs impacts sur la performance de routage ad hoc. En se basant sur des

études précédentes et nos propres simulations, nous avons déduit que les attaques par suppression des paquets constituent une menace potentielle contre la disponibilité du routage ad hoc. Nous avons discuté certains mécanismes de sécurité qui ont été proposés pour lutter contre ces attaques. Nous avons donné plus d'importance aux systèmes de détection, précisément ceux qui sont basés sur des modèles de classification.
Nous avons proposé un mécanisme de détection basé sur une analyse comportementale et utilisant deux modèles Bayésiens pour la classification des nœuds qui sont : Bernoulli et Multinomial. Nous avons fait une description des différentes phases de détection nécessaires pour surveiller et modéliser les comportements d'un nœud, et après les classifier selon la valeur de leur probabilité de malveillance. Les résultats de simulations ont montré que les modèles Bayésiens peuvent être utilisés pour assurer la détection d'attaques DoS par suppression des paquets.
Nous avons exploité l'approche proposée dans la première contribution pour proposer un mécanisme de prédiction des comportements des nœud basé sur les chaînes de Markov. Nous avons montré que les nœuds réalisant des attaques par suppression périodique des paquets peuvent être détectés en effectuant une traçabilité de leurs comportements. Nous avons défini un ensemble de trois niveaux de malveillance représentant l'espace d'états des comportements en utilisant un modèle basé sur la logique floue. Nous avons présenté une analyse stochastique permettant de modéliser l'évolution de l'état d'un nœud sous forme d'une matrice de transitions. Les entrées de cette matrice sont des valeurs de probabilités de transitions entre trois états des comportements : légitime, suspect et malveillant. Nous avons montré via simulations la capacité de notre solution de prédire l'état limite des nœuds en se basant sur l'évolution de leurs comportements. D'autre part, nous avons montré que les attaques par suppression périodique des paquets peuvent être détectées avec un taux minimal de fausses alertes.

Nouveaux défis émergents

L'objectif de solutions proposées dans cette thèse est d'adresser le problème de disponibilité de routage ad hoc. Les mécanismes proposés sont fondamentalement complémentaires et conçus pour assurer une détection complète des différentes types d'attaques par suppression des paquets. Cependant, certains défis techniques sont identifiés suite à notre évaluation globale de nos solutions, surtout en ce qui concerne la modélisation des comportements et la classification probabiliste des nœuds.
Dans les modèles de classification utilisés, le calcul de taux de transmissions des paquets des données s'appuie sur deux types de paquets de contrôle (RREQ et RREP) comme des attributs. Cette sélection est basée sur le fait que les comportements d'un nœud peuvent être bien présentés en évaluant sa participation dans les services d'acheminement de données, découverte des routes et établissement des routes. Nous avons omis les deux autres types des paquets de contrôle utilisé dans le protocole AODV qui sont RERR et Hello, en assumant que leur suppression n'a pas un effet significatif sur la performance de routage. Suite à nos études récentes, nous avons

constaté que la suppression et la manipulation de ces paquets peuvent perturber la disponibilité de services de routage.
Les messages Hello sont utilisés normalement pour maintenir les liens entre les nœuds qui sont à un saut l'un de l'autre. Ces messages sont des paquets RREP avec un nombre de saut égale à 1, qui sont utilisés pour envoyer une information de présence sans aucun rôle ultérieur. Donc, la suppression de ces paquets par un nœud récepteur n'a pas aucun effet négatif sur la performance de routage. Cependant, l'altération des messages Hello peut être exploitée par un nœud malveillant pour réaliser des attaques de détournement de trafic, ce qui est actuellement hors de notre objectif.
D'autre part, les messages RERR sont utilisés pour informer les nœuds de l'occurrence d'une rupture de lien. Ce message est envoyé par le nœud qui a détecté la rupture vers les nœuds qui utilisent ce lien pour envoyer leurs paquets. Si un nœud malveillant supprime un message RERR qui est supposé d'être transmis, les nœuds assumés d'être informés de cette rupture seraient susceptibles de continuer à utiliser le lien rompu pour envoyer leurs paquets. Par conséquent, les paquets des données passant par cette lien peuvent être perdus, ce qui peut perturber la performance de routage.
Dans les deux solutions présentées dans les chapitres 4 et 5, le nœud surveillant se base sur des observations directes pour décider si un autre nœud surveillé est malveillant ou non. Cependant, dans la plupart des solutions utilisant les systèmes de confiance, la décision sur un nœud surveillé est basée sur une agrégation des observations directes et les recommandations des autres nœuds. Dans ce cas, rien n'empêche un nœud malveillant de générer des mauvaises recommandations pour tromper le mécanisme d'évaluation de confiance [48]. Par conséquent, nos solutions sont moins vulnérables à ce type de comportements malveillants puisque la décision sur un nœud est prise sans faisant référence aux recommandations d'autres nœuds. Cependant, cette manière de prise de décision peut engendrer des fausses alertes quand la détection des nœuds malveillants est réalisée dans un réseau dense, où il est difficile de décider si la suppression des paquets est dû à une attaque ou une collision par exemple. Dans ce cas, l'agrégation des observations directes et indirectes semble d'être efficace pour mitiger telles fautes de décisions. Un nœud surveillé qui est vu comme malveillant par un nœud surveillant peut être considéré comme légitime de point de vue des autres nœuds. Par conséquent, l'utilisation des recommandations comme une information de seconde main peut diminuer le taux de fausses alertes dans telles situations.
Finalement, la proposition d'une solution qui prend en considération les limitations de ressources n'était pas un objectif prioritaire dans cette thèse. La complexité des opérations de surveillance, la modélisation des comportements et les algorithmes de classification n'ont été traités ni évalués. En plus, l'espace mémoire et le surcharge réseau nécessaire pour faire la traçabilité des nœuds n'étaient pas abordés. Puisque la disponibilité de services de routage est notre objectif prioritaire, nous planifions de prendre plus de considération aux contraintes des ressources.

Travaux de recherche en cours

Pour traiter le problème d'attaques menaçant la disponibilité de routage, nous avons réalisé deux travaux de recherche qui proposent des solutions pour les attaques DoS par épuisement des ressources (Flooding) et par manipulation de trafic (Wormhole). Nous avons étudié les techniques utilisées par ces attaques et interprété leurs effets sur la performance de routage.
Nous avons proposé une solution préliminaire contre les attaques Wormhole en utilisant un modèle basé sur la théorie des graphes. Le mécanisme de détection proposée est déclenchée par un nœud destination quand il reçoit un message de demande de route (RREQ), et avant envoyer un message de réponse de route (RREP) vers la source de demande. Ce mécanisme consiste à vérifier si ce message RREQ a été diffusé par un nœud suspect en comparant le nombre de sauts traversés par ce message avec celui des autres nœuds qui ont encore diffusé le même message RREQ. Ce travail a été publié dans une conférence nationale ; les lecteurs intéressés peuvent découvrir plus des détails dans [49].
D'autre part, nous avons étudié les attaques d'épuisement de ressources par inondation RREQ qui menacent le protocole de routage AODV. Nous avons conçu une solution pour permettre à un nœud de détecter telles attaques en utilisant une méthode statistique. L'idée de base consiste à surveiller et évaluer la moyenne mobile pondérée des messages RREQ générés par un nœud pour détecter les générations anormales des demandes de route. Ce travail a été publié dans une conférence internationale et plus détails sur la solution proposée peuvent être trouvés dans [50].

Perspectives

Le travail réalisé dans cette thèse nous a motivé à élaborer certains travaux de recherche à court terme comme suivant :

- **Adaptabilité des solutions**: les solutions que nous avons proposé dans les chapitres 4 et 5 nécessitent plus d'études en termes de précision de détection. Une analyse ultérieure de la phase de monitorage est nécessaire en ce qui concerne les capacités matérielles des nœuds d'utiliser le mode promiscuité. D'autre part, il est important de raffiner les critères de détection des attaques par suppression des paquets en interprétant l'impact de choix des seuils sur la précision de détection.

- **Mobilité des nœuds**: le mécanisme de prédiction présenté dans le chapitre 5 nécessite plus d'analyse en termes de traçabilité des nœuds. La phase de monitorage doit être analysée en particulier dans le cas des réseaux à forte mobilité .

- **Incertitude de détection**: selon les résultats des simulations obtenus dans le chapitre 5, un pourcentage de 5% des nœuds non-détectés persiste même en augmentant la durée de monitorage. Nous allons étudier l'impact de configu-

rations des seuils et de la durée de monitorage sur le pourcentage des nœuds non-détectés.

- **Décisions de routage**: dans le chapitre 5, la distribution limite des probabilités définissant l'état stationnaire d'un nœud est utilisée pour identifier la nature des ses comportements. Ce process nécessite plus de spécifications pour augmenter la précision et mitiger l'incertitude des décisions prises en un nœud.

Nous soulignons quelques directions de recherche que nous allons envisager à long terme :

- **Analyse de complexité**: les techniques statistiques et probabilistes présentés dans les chapitres 4 et 5 doivent être analysées et améliorées en termes de complexité. Nous pouvons évaluer la performance de nos solutions dans un réseau réel et optimiser les différentes phases de détection pour minimiser leur surcharge sur le réseau.

- **Contraintes des ressources**: l'espace mémoire et l'énergie nécessaires pour stocker et traiter les informations collectés sur les comportements et l'évolution d'un nœud méritent plus d'analyse en termes de consommation des ressources. Ce dernier peut être étudié dans le cas d'une implémentation réelle pour concevoir une solution adéquate contre les attaques de suppression des paquets en tenant compte les ressources nécessaires pour assurer les différentes phases de détection.

- **Implémentation réelle**: les solutions proposées dans cette thèse sont capable de détecter les différentes types d'attaques par suppression des paquets. Cependant, il est nécessaire d'évaluer leur adaptabilité d'être implémentées comme un système de détection au niveau de chaque nœud du réseau. Nous pouvons encore adapter ces solutions comme une extension de sécurité qui agit d'une manière similaire aux systèmes de réputation.

- **Suppression coopérative des paquets**: dans cette thèse nous avons traité le cas des nœuds malveillants qui réalisent des attaques par suppression sans aucune coopération entre eux. En se basant sur des études récentes, les attaques par suppression des paquets qui sont réalisées par plusieurs nœuds d'une manière coopérative peuvent rester dans le réseau et perturber le routage sans être détectés. Nous allons traiter le problème de ces attaques dans les réseaux MANETs en analysant leurs techniques et essayant de modéliser leurs comportements.

Abstract

With the evolution of user requirements, many network technologies have been developed based on the Machine-to-Machine (M2M) communication concept. Among these technologies, we find Mobile Ad hoc Networks (MANETs) that were designed to ensure communication in situations where the deployment of a network infrastructure is expensive or inappropriate. In this type of networks, routing is an important function where each node acts as a router and participates in routing services.
Basically, MANET entities are in a managed environment where only authorized users can participate in the network. However, some ad hoc scenarios are in an open environment where nodes come from different organizations or places and do not know each other in advance. Moreover, existing routing protocols are not designed with security in mind and are often vulnerable to attacks performed by malicious entities. For instance, an authenticated entity may behave maliciously by dropping the received packets that are supposed to be forwarded, in the aim of disrupting the routing services and blocking the network traffic.
In this thesis, we first present a taxonomy of ad hoc routing attacks, precisely those leading to a denial of routing services. The main characteristic of this work is that it distinguishes different objectives and mechanisms of the Denial of Service (DoS) attacks, which can help defenders to easily notice which attacks should be prevented. We then focus on our main research objective which is the proposition of a fully distributed detection mechanism of malicious nodes performing packet dropping attacks to disrupt the routing services in MANETs.
We propose at first a classification framework based on a Bayesian probabilistic analysis in order to evaluate the behavior of a node based on its interaction with its neighbors using a completely decentralized scheme. Simulation results show that misbehaving nodes can be efficiently detected using the Bayesian classifiers.
Besides, we propose a prediction framework extending the detection mechanism already mentioned using a Markov chain model to handle the problem of periodic packet dropping attacks. The core idea of this approach consists of keeping track of the evolution of network nodes over a time period in order to predict their stationary states. Simulation results show that the proposed solution is able to predict the state of a node based on its historical evolution and then detect the periodic dropping attacks with an accuracy rate greater than 90%. Finally, we use the experience obtained in this thesis to provide some guidelines for security enhancements that should be considered to guarantee the availability of routing services in MANETs.

List of Figures

List of Tables

Glossary

3G	Third generation of mobile telecommunications technology
ABM	Anti-Blackhole Mechanism
AE	Adaptive Encryption
AF-AODV	Anti-Flooding AODV
AODV	Ad hoc On-demand Distance Vector
AOMDV	Ad-hoc On-demand Multipath Distance Vector
AP	Access Point
ARAN	Authenticated Routing for Ad hoc Networks
ASR	Anonymous Secure Routing
BS	Base Station
CA	Certificate Authority
CBR	Constant Bit Rate
CFR	Control packet Forwarding Ratio
CH	Cluster Head
CONFIDANT	Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks
CORE	COoperative REputation

CPU	Central Processing Unit
CTAC	Control Traffic Tunneling Attack Countermeasure
CTS	Clear-To-Send
CUSUM	CUmulative SUM
DDoS	Distributed Denial of Service
DDWS	Dual Defensive Wall System
DelPHI	DELay Per Hop Indication
Dest-Seq	Destination Sequence Number
DFR	Data packet Forwarding Ratio
DoS	Denial of Service
DPRAODV	Detection, Prevention and Reactive AODV
DRI	Data Routing Information
DSDV	Destination-Sequenced Distance Vector routing
DSR	Dynamic Source Routing
ETX	EXpected Transmission count
FFT	Fast Fourier Transform
FIFO	First In First Out
FN	False Negative
FP	False Positive
FPNT-OLSR	Fuzzy Petri Net based Trust OLSR
FrAODV	Friendship based AODV
FSM	Finite Machine State
FWAD	Frequency-based Wormhole Attack Detection

GPS	Global Position System
GTK	Group Transient Key
HSRBH	Hierarchical Secure Routing against BlackHole
ID	IDentifier
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IN	Intermediate Node
IoT	Internet of Things
IP	Internet Protocol
LTE	Long Term Evolution
M2M	Machine-to-Machine
MAC	Media Access Control
MAD	Mutual Authentication Distance-Bounding
MANET	Mobile Ad hoc NETwork
MCE	Mass Casualty Event
MPR	MultiPoint Relay
N/A	Not Applicable
NFC	Near Field Communication
NHN	Next Hop Neighbor
NNT	Neighbor Number Test
NS2	Network Simulator 2

OLSR	Optimized Link State Routing
P2P	Peer-to-Peer
PDA	Personal Digital Assistant
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PTK	Pairwise Transient Key
QoS	Quality of Service
RAD	Random Assessment Delay
RDP	Route Discovery Packet
REP	REPly
RERR	Route ERRor
RFC	Requests For Comments
RFID	Radio Frequency IDentification
RREP	Route REPly
RREQ	Route REQuest
RRT	Receiving Record Table
RTS	Request-To-Send
RTT	Round-Trip Time
SAM	Statistical Analysis of Multi-path routing
SAODV	Secure AODV
SEAODV	Secure Efficient AODV
SECTOR	SECure Tracking Of node encounteRs

SHARP	Sharp Hybrid Adaptive Routing Protocol
SIDE	Specification-based Intrusion DEtection
SLR	Short Retry Limit
SN	Sequence Number
SPAN	Smart Personal Area NETwork
SRP	Secure Remote Password
SRT	Self Record Table
SWAN	Statistical Wormhole Apprehension using Neighbors
TAODV	Trusted AODV
TC	Topology Control
TCP	Transmission Control Protocol
TCR	Total Cost Ratio
TE	Triangular Encryption
TESLA	Time Efficient Stream Loss-tolerant Authentication
TIK	Tesla with Instant Key disclosure
TN	True Negative
TP	True Positive
TQR	Trust-based QoS Routing
TREP	Trust REPly message
TREQ	Trust REQuest message
TSR	Trust-based Source Routing protocol
TTL	Time to Live
TTM	Transmission Time-based Mechanism
TWARN	Trust Warning message
UDP	User Datagram Protocol

V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANET	Vehicular Ad hoc NETwork
VoIP	Voice over Internet Protocol
WBAN	Wireless Body Area Network
WiFi	Wireless Fidelity
WiMax	Worldwide Interoperability for Microwave Access
WMN	Wireless Mesh Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
ZRP	Zone Routing Protocol

Chapter 1

Introduction

1.1 Background and motivation

Over the last decade, the world has witnessed important advances in mobile computing along with the development of wireless technologies, enabling information and service access anywhere, anytime and from any device. In addition, mobile communications played a central role in improving quality of life by connecting people, homes, cars and other social systems in the context of so-called Smart city [1]. Today, wireless mobile devices are essential in the daily life of millions of users, who keep their laptops, smart phones and tablets nearby, check them frequently and sometimes use multiple devices at once. These devices, even if they are heterogeneous in terms of characteristics and goals, can share their resources in a network and constitute a Mobile Ad hoc Network (MANET).
MANETs are autonomous systems of wireless mobile nodes that can be dynamically self-organized into an arbitrary and temporary network topology [2]. MANETs do not rely on a predefined infrastructure; every node in the network should act as host and a router at the same time. MANETs can provide important means of achieving the ubiquitous network utilization thanks to their ease, low cost and robust deployment. These networks are recommended for providing temporary communications in highly dynamic and harsh environments such as, battle fields, disaster relief and rescue operations.
Despite the fact that infrastructure-less feature of MANETs offers a wide variety of deployment facilities, it makes the network vulnerable to security challenges, especially in open environments where nodes are heterogeneous and do not know each other in advance. In such challenging context, more responsibilities are in charge of network nodes, which should cooperate with each other and provide different network services. However, existing ad hoc routing protocols do not support the realization of these intrinsic responsibilities even if they are cooperative in nature. In fact, most of them are not designed with security in mind, since they operate

under a trust assumption on all network nodes. Therefore, many security attacks can take place against the network especially at the routing layer. For instance, a malicious node may discard some or all packets it receives to deny their forwarding towards the intended destination. On the other hand, being wireless enabled small devices having limited resource in terms of battery life and bandwidth, MANET nodes trend to be selfish. Hence, it is possible for a node to decide not to cooperate in packet forwarding in order to save its resources as long as possible.
Security is a fundamental requirement in MANETs which guarantees the prevention of those malicious behaviors that may intercept, modify and discard packets, or inject incorrect topological information to disrupt routing services. However, security in MANETs cannot be guaranteed through centralized mechanisms used in infrastructure network such as, firewalls and network-based Intrusion Detection Systems (IDS).
A secure routing in MANETs should guarantee the *integrity* and *privacy* of routing information, the *authentication* and *non-repudiation* of network nodes and the *availability* of network services [3]. This latter means that network resources and services should remain available even in presence of malicious nodes in the network, which can be ensured by preventing service disruptions caused by Denial of Service (DoS) attacks [51].
At the network layer, *availability attacks*, the alternative name of DoS attacks, can be classified in three main categories:

- *Resource consumption attacks*: can be launched by an external attacker by generating an excessive number of bogus packets in the aim of exhausting the resources of network nodes.

- −]*Routing disruption attacks*: exploit protocol vulnerabilities to attract network traffic or establish malicious links between colluding nodes. Further, the intercepted traffic may be altered or discarded depending on attacker goals.

- *Packet dropping attacks*: in this class of attacks, when a malicious node receives packets that are supposed to be forwarded, it discards some or all of them in the aim of disrupting their forwarding to an intended destination.

Due to their severe impact on network performance, especially on the routing services availability, packet dropping attacks have attracted a considerable attention of many research works. *Cryptography-based* mechanisms were widely proposed as preventive solutions against these attacks. In general, these mechanisms were proposed to authenticate network nodes [18][17] and ensure the integrity of routing information using public key cryptography [5] or symmetric key cryptography [20][21]. Other research works showed that *trust management* is an essential requirement in MANETs, especially when nodes desire to establish a network with a certain level of trust relationships among themselves and without any previous interactions [6]. Trust-based security mechanisms were generally used to force selfish nodes to cooperate [23], isolate them from the network [24] or establish a

reputation system to make routing decisions based on trust level of nodes [7] [26] [52].

Based on previous studies [8], we consider that the cryptographic measures alone are not sufficient to counteract different security problems caused by malicious nodes. On the other hand, we believe that trust management in MANETs is challenging due to the topology changes which may hamper the trustworthiness evaluation and aggregation among network nodes.
Therefore, we aim to propose a fully decentralized mechanism to overcome the challenges that face most of cryptographic and trust-based solutions. We investigate statistical and probabilistic techniques to design and implement a novel behavior-based detection mechanism against dropping attacks in MANETs. We believe that such a solution faces a range of challenges:

- **Decentralization**: the lack of centralized management and security enforcement points in MANETs such as, routers and firewalls complicate the detection of misbehaving nodes. How can we ensure that all detection phases are *fully decentralized*?
- **Node mobility**: the unpredictable mobility of network nodes causing frequent topology changes affect the detection accuracy of misbehaving nodes. How can we design a detection mechanism which can be adapted to the *dynamic* nature of MANETs?
- **Resource constraints**: MANET nodes have the tendency to be selfish and attempt to preserve their limited resources especially in terms of bandwidth and battery power. What are the characteristics of a *lightweight* detection mechanism in MANETs?
- **Detection accuracy**: the detection should define a set of attributes in order to realize a behavior-based analysis of network nodes and then detect those misbehaving ones. How can we design a *model of nodes' behaviors* that ensures an accurate detection of malicious nodes?
- **Node's behavior changes**: a sophisticated attacker may perform a selective packet dropping or change its behavior over time in order to deceive the underlying detection mechanism. How can we exploit the *evolution* of a node during a time period to identify the nature of its behaviors?

1.2 Contributions

The first contribution in this thesis consists of proposing a novel taxonomy of availability attacks threatening the ad hoc routing layer. The major attack scenarios that can actively lead to a denial of routing services can be classified in three broad categories: packet dropping, resource consumption and routing disruption. For each class of attacks, we investigate their techniques, objectives and damages they cause to the ad hoc routing services.

The second contribution consists of a decentralized detection mechanism against dropping attacks based on a probabilistic classification of network nodes. This mechanism is designed to allow a node to recognize the behaviors of a neighbor node before forwarding packet through it. We exploit the ability of MANETs to work in a *promiscuous mode* to ensure the monitoring of all packets in a node's transmission range regardless their destination [9]. We model node's behaviors as a vector based on three attributes representing the forwarding rates of three packet types: Data packets, RREQ and RREP routing messages. Based on attribute values, we perform a classification of a behaviors' vector by evaluating its probability of maliciousness using two *Bayesian models*: Bernoulli and Multinomial. We use the *Total Cost Ratio* (TCR) to evaluate the performance of these classifiers by comparing their error rate with that of the baseline classifier. Simulation results show that the use of Bayesian classifiers ensure an accurate detection of malicious nodes in MANETs. Bernoulli classifier shows a full detection of malicious nodes if their proportion in the network exceeds 15%, while Multinomial classifier is more appropriate when the proportion of malicious nodes is less than 27%. We note that the combination of these classifiers can ensure a full detection of packet dropping attacks in MANETs.

The third contribution is an extension of the first approach, in which we propose to track the evolution of a node's behaviors by calculating its probability of maliciousness at each time slot τ during a time period T. We use a *fuzzy logic* model to associate for each obtained probability value a linguistic label representing the node's behaviors which are detected at a time slot τ. We define three levels of maliciousness representing the space of possible behavior states of a node over T: *legitimate*, *suspicious* and *malicious*. We use the sequence of transition between these states to represent the evolution of a node as a stochastic matrix using a *Markov chain*. We show through simulation the ability of our proposed solution to predict the state of a node by applying the *ergodicity theory* with a number of steps less than 10. We clarify the important role of the suspicious state that we introduce to overcome the challenge of the periodic packet discarding and then increase both QoS and security. Simulation results show that our solution is able to detect different types of dropping attacks with an accuracy rate greater than 90% in different configuration parameters in terms of: network size, percentage of malicious nodes and simulation time.

1.3 Dissertation outline

The thesis is organized in six chapters which are further divided into multiple sections. Hereafter, an overview of those chapters is provided:

Chapter 1, "Introduction", also the current chapter, provides a global view of the thesis. It introduces the context in which our research work is realized, the importance of MANETs, the security threats at the ad hoc routing layer and an overview of existing security solutions. It focuses on the problem of packet dropping attacks and **presents the technical challenges that should be taken into consideration to design an adequate misbehavior detection mechanism**. It also presents the motivations, the contributions and the organization of this thesis.

Chapter 2, "MANETs: applications, protocols and security issues", provides firstly an overview of ad hoc communication technologies and their applications in the context of Smart city, and then **studies the routing characteristics, protocols and security threats in MANETs**. It provides a taxonomy of DoS attacks threatening the availability of ad hoc routing services, and focuses on the techniques they used to realize their malicious objectives. Finally, it summarizes the security vulnerabilities in ad hoc routing protocols that can be exploited by a malicious entity to perform a DoS attack.

Chapter 3, "Security mechanisms against DoS attacks in MANETs", **surveys the solutions that were proposed in the literature to secure routing services and protocols in MANETs**. In this chapter, we analyze security mechanisms for each of three classes of DoS attacks: packet dropping, resource consumption and routing disruption.

Chapter 4, "A probabilistic detection mechanism against routing misbehaviors in MANETs", proposes a decentralized solution to detect packet dropping attacks at the ad hoc routing layer. In this chapter, we first describe the network assumptions, the modeling scheme of nodes' behaviors and the specification of the proposed detection mechanism. We also present the Bayesian models used to **evaluate the probability of maliciousness of nodes' behaviors**. Finally, we evaluate the performance of the proposed solution based on results that we obtain through network simulations.

Chapter 5, "Nodes' behaviors prediction through a stochastic analysis", presents an extension of the detection mechanism proposed in chapter 4. In this chapter, we address the problem of periodic dropping attacks in MANETs. We use a fuzzy logic model to associate to a probability of maliciousness value a linguistic label defining the state of its behaviors. We propose a stochastic Markov chain to model the state evolution as a stochastic transition matrix. We show through simulations the ability of the proposed solution to **predict the state of a node based on its**

previous state transitions. In addition, we prove through simulation that different types of packet dropping attacks can be detected with a high rate of accuracy.

Chapter 6, "Conclusion and future directions" concludes the dissertation with a review of the work that we realized in this thesis, a global evaluation of the proposed solutions and an investigation of emergent technical challenges. It also **presents an overview of other research contributions that we realized to detect *Flooding attacks* and *Wormhole attacks*** that threat the availability of ad hoc routing services. Finally, it presents an overview of our short-term and long-term future research directions.

Chapter 2

MANETs: applications, protocols and security issues

"To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science."

– *Albert Einstein*

Contents

2.1 Introduction

Recent advances in mobile computing along with the development of mobile devices enable information and service access anywhere, anytime from any device.

With the evolution of user requirements, many network technologies were developed based on the concept of *Machine-to-Machine* (M2M) communication. Among these technologies, *Mobile Ad hoc Networks* (MANETs) have been designed to ensure communications where the deployment of a network infrastructure is expensive or inconvenient. Being an infrastructure-less network, MANET's nodes ensure network service cooperatively without relying on a central administration. The limited transmission power of nodes implies that communications beyond the radio range must rely on the forwarding help of other nodes.
Due to their intrinsic characteristics, MANETs are exposed to a large number of security threats besides those inherited from conventional wireless networks, especially at the routing layer. Studying threats against routing layer is a fundamental requirement to detect the potential attacks, and then build an adapted security architecture for specific ad hoc applications. The major security goals in MANETs consist of providing the following security attributes: availability, confidentiality, integrity, authentication and non-repudiation.
In this context, this chapter aims to focus mainly on application and security challenges in MANETs, and is organized as follows: fundamental concepts of ad hoc communication technologies, routing protocols, applications and limitations are introduced in section 2.2. In section 2.3, we present an in-depth security analysis of *Denial of Service* (DoS) attacks at the ad hoc routing layer. We discuss the protocol vulnerabilities and attack techniques exploited by attackers and the techniques targeting the availability of routing services in MANETs.

2.2 Background: Ad hoc networks

Ad hoc networks are defined as wireless networks that do not rely on any predefined infrastructure. Thanks to these characteristics, ad hoc networks have a wide range of potential applications. In this section we provide an overview of different applications and real deployments of this type of networks. On the other hand, we show the characteristics of the ad hoc routing and review some specific routing protocols.

2.2.1 Applications of ad hoc networks

Over the last decade, smart phones and tablets have emerged as a multi-purpose computing platform, relying exclusively on wireless connectivity and replacing personal digital assistants and low-end mobile computers. In this context, the ad hoc concept played a key role to ensure the communication between these devices.
Many research works have studied different issues related to ad hoc networks such as, routing protocols, quality of service and security challenges. However, there are a few research works on current and future deployments of ad hoc communications. According to recent statistics provided by Cisco VNI Service Adoption Forecast in figure 2.1, we note a significant increase in M2M communications during the last four years decade, and more increasing in the two coming years.

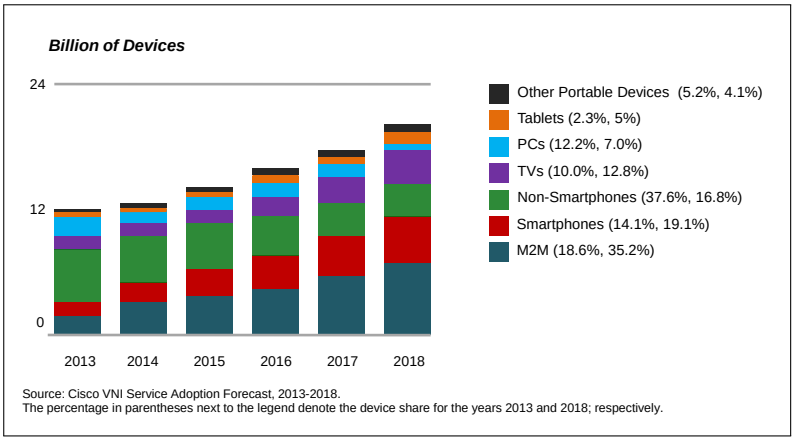


Figure 2.1: M2M communications increase.
Source: www.cisco.com

In the following, we briefly survey some applications of M2M communications and areas of interest for ad hoc networks, especially in the context of *smart city* and *Internet of Things* (IoT).

2.2.1.1 Vehicular networking

This area covers applications where one of the communication partners is a vehicle. *Vehicular ad hoc networks* (VANETs) aim to ensure information exchange between vehicles and provide several types of network services [10].

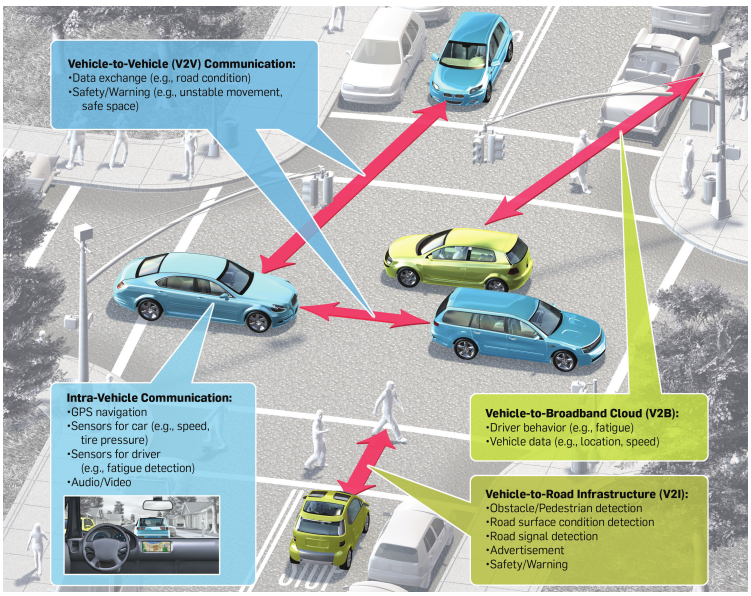


Figure 2.2: An overview of vehicular networking areas.
Source: www.decom.ufop.br

According to figure 2.2 there two main types of communications used in VANETs. The first communication mode of VANETs is *Vehicle-to-Vehicle* (V2V) communication, which has applications in convoy driving, lane changes and other safety services. For instance, when a vehicle detects a new event (e.g., accident), it notifies neighboring vehicles about traffic conditions change and helping drivers in decision making.
The second communication mode is called *Vehicle-to-Infrastructure* (V2I) communication. In V2I, the infrastructure plays a coordination role by gathering global or local information on traffic and road conditions and then suggesting or imposing certain behaviors on a group of vehicles. One example is ramp metering, already widely used, which requires limited sensors and actuators. Finally, VANETs ensure some comfort services by offering not only an Internet access, but also guiding drivers to find available spaces in a near parking [53]. VANETs also include *Vehicle-to-Broadband Cloud* communication (V2B), which involves generally the short range communication between the vehicle and personal devices carried by passengers using different technologies, such as *3G* cellular telephony [54], *Bluetooth* [55], *WiMax* [56] and *LTE* [57].

2.2.1.2 Urban sensing

The focus of wireless sensor networking research has evolved from static networks of specialized devices to advanced network technologies making use of robotic or other controlled mobility to adapt to the sensing conditions and a people-centric approach relying on the mobility of people [11].

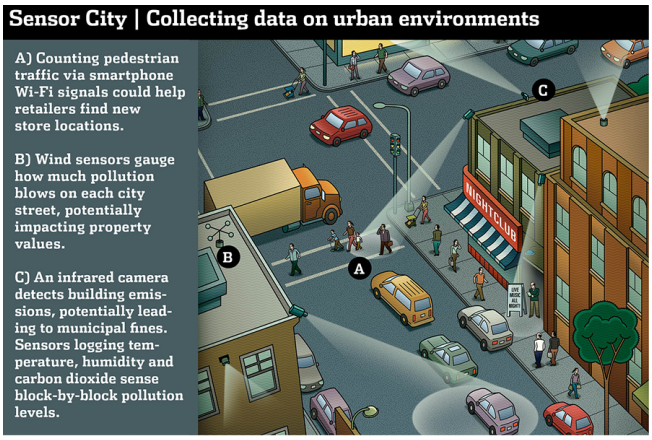


Figure 2.3: Scenario examples of sensing process in urban environments.
Source: www.novim.org

Initiatives in the area of *Wireless Sensor Networks* (WSN) propose on one level to make infrastructures more efficient, but on another level, citizens with sensing capabilities are a key way in which urban processes may run more efficiently by

monitoring their individual consumption activities, transport patterns and energy use (see figure 2.3). An expanding research community is developing techniques to bring about very large scale urban sensing by leveraging the increasing sensing capabilities found in consumer devices such as smartphones. Data collected from these mobile sensors provide the foundation for exciting people-centric applications and projects such as, *Google Street View* [58], *MIT Senseable City Lab* [59] and *Intel Urban Atmospheres project* [60].

2.2.1.3 Ubiquitous computing

Ubiquitous, also called pervasive, means "existing everywhere" [14]. In contrast to desktop computing, ubiquitous computing can occur using any device, in any location and in any format. This paradigm is the result of computer technology advancing at exponential speeds, with the aim of integrating computation into the environment.
At their core, all models of ubiquitous computing share a vision of small processing devices, distributed at all scales throughout everyday life. For example, a *smart house* environment might interconnect heating controls with personal biometric monitors woven into clothing so that environment conditions in a room might be modulated continuously and imperceptibly [61].
Another example involves solutions where a television set-top box can be controlled from a smartphone, through an Internet connection, even when the two devices are several feet from each other.

2.2.1.4 Network extension

In this application area, ad hoc networks are used to improve the performance of networks having an insufficient coverage [12]. Adding an ad hoc extension to the access network will benefit both service providers and users. In that case, while some nodes access the Internet by directly connecting to an *Access Point* (AP), others might be sending and receiving data packets through those intermediate nodes to have Internet access.
In this context, *Wireless Mesh Networks* (WMNs) are the most known to ensure network extension and Internet access solution [62]. These networks have the potential to offer low cost, wireless broadband Internet access for both fixed and mobile users [63]. In WMNs, nodes can communicate directly with each other, without requiring the assistance of an Internet connection. Therefore, if one node can no longer operate, other nodes can still communicate with each other, directly or through one or more intermediate nodes. WMNs allow a scalable coverage, high fault tolerance and low installation costs, which can be exploited to increase the coverage area of network services.

2.2.1.5 Wireless Body Area Network (WBAN)

The *body area network* field is an interdisciplinary area which can allow inexpensive and continuous health monitoring with *real-time* updates of medical records through the Internet.
Wearable health monitoring systems integrated into a *tele-health system* are novel information technology that will be able to support early detection of abnormal conditions and prevention of its serious consequences [15].

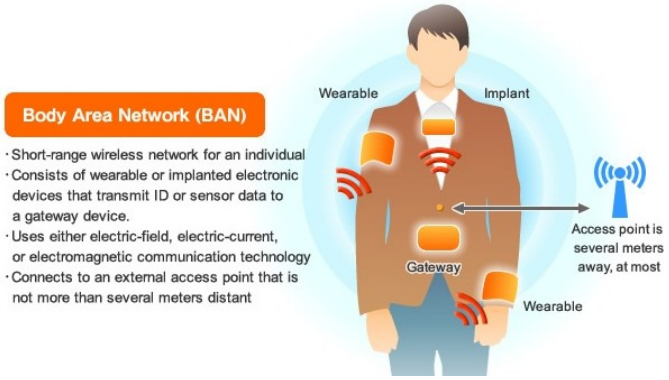


Figure 2.4: WBAN of Intelligent Sensors for Patient Monitoring.
Source: www.rs-online.com

Wearable sensor nodes can store patient data such as, identification, history and treatments, by supplementing the use of back-end storage systems and paper charts (see figure 2.4). In a *Mass Casualty Event* (MCE), these networks can greatly improve the ability of first responders to treat multiple patients equipped with wearable wireless monitors [64].

2.2.1.6 Wireless Personal Area Network (WPAN)

Ad hoc networks allow a number of independent data devices to communicate without any central administration. Communications in WPANs are normally confined to a person or object that typically extends up to 10 meters in all directions and envelops two or more objects or persons whether stationary or in motion [13]. WPANs could serve to interconnect all communicating nodes that many people have on their desk or carry with them today.
In the context of personal utilization, *Smart Phone Ad hoc Networks* (SPANs) leverage the existing hardware in commercially available smartphones to create *Peer-To-Peer* (P2P) networks without relying on cellular carrier networks or wireless access points. This technology differs from traditional hub and spoke networks such as, WiFi Direct [65], in that they support multi-hop relays, so peers can join and leave at anytime without destroying the network.
Another paradigm of WPANs applications is the *Near Field Communication* (NFC)

system, which allows two NFC-enabled devices to communicate with each other and exchange information in an ad hoc fashion [66]. This contact-less technology allows devices to communicate among themselves by a simple tap or touch to make use of many services such as, cashless payment, P2P data transfer, loyalty and membership identification and other real time applications.

2.2.2 Summary

The numerous deployments of ad hoc networks, especially in smart city deployments and IoT applications, make them an essential component of future Internet architecture. Routing is one of the services that will undergo a notable evolution, where traditional routers will be replaced by smart entities.
Networks will have more distributed operations; a huge size of data will be stored and forwarded daily through users' devices. Hence, there will be a crucial need for an appropriate routing service to overcome the evolution non-predictable [67]. In this context, ad hoc routing protocols constitute an important topic when designing reliable and secure communications over distributed systems and smart applications. In the following, we introduce the ad hoc routing concepts, characteristics and protocols.

2.2.3 Fundamental concepts of ad hoc routing

The routing function constitutes the basis for data exchange, which enables the establishment and maintenance of optimal routes between network nodes. In conventional wireless networks, routing protocols do not need to manage constraints related to mobility nor topology changes, since the routing function is ensured by fixed entities (i.e., routers).

2.2.3.1 Routing protocols classifications

Being an infrastructure-less, dynamic and resource constrained networks, MANETs require a dedicated routing protocol which satisfies the following requirements:

- Distributed way of route establishment between nodes.
- Adaptation to frequent topology changes by handling broken links at real time.
- Low overhead as well as a low consumption of energy.

Many routing protocols were proposed and dedicated for ad hoc routing, where each protocol defines its own metrics to ensure an optimal route selection. Some protocols use hop count as a selection metric, and then choose the shortest path between communicating nodes [68]. Other protocols take into account other metrics, such as *Quality of Service* (QoS) [69], reliability [70] or security considerations [71]. According to how routing information is structured and exchanged, ad hoc routing protocols can be classified in two classes of approaches:

- The *distance vector* approach, where routing information is only exchanged between directly connected neighbors.
- The *link state* approach, which requires that all routers know about the paths reachable by all other nodes in the network.

Ad hoc routing protocols can be also classified in three types according to how the routes are discovered and updated [72]. They are respectively proactive, reactive and hybrid protocols.

- **Proactive:** also called "*table-driven*". Nodes periodically exchange messages in order to overcome the dynamic aspect of network topology and ensure the consistence of routing information. Proactive approaches allow an optimized route discovery; a route already discovered can be used instead of constantly search again.
- **Reactive:** also called "*on-demand*". Nodes do not exchange routing information until there is data to be sent but no route is available. To find out a route, a sender node broadcasts a route request message to its neighbors hoping it will reach the requested destination. A route reply message is sent once this request message reaches the destination.
- **Hybrid:** this class of routing protocols aims to combine the merits of both proactive and reactive schemes. Each node maintains the topology information within its coverage area using a proactive approach. The routes outside the node's coverage area are discovered using a reactive approach, as is the case with the *Zone Routing Protocol* (ZRP) [73], and the *Sharp Hybrid Adaptive Routing Protocol* (SHARP) [74].

In the following, we introduce two representative ad hoc routing protocols, *Ad hoc On-demand Distance Vector protocol* (AODV) [16] and *Optimized Link State Routing protocol* (OLSR) [75]. For the other routing protocols, interested readers can refer to [12] for more information.

2.2.3.2 The AODV routing protocol

This reactive protocol was developed in Nokia Research Center, University of California, Santa Barbara and University of Cincinnati [16]. The key concept of AODV is to establish routes only when needed. This protocol has two main phases: *route discovery* and *route maintenance*.

Route discovery: this process is triggered once a source S wants to transmit data to an unknown node D, and that S has no fresh route to D. S broadcasts a *Route Request* (RREQ) message to its neighbors to ask them if they have a route to D. At an intermediate node I, if the RREQ is received for the first time, a reverse route towards the source node is created. Otherwise, the packet is discarded by the node I. When receiving a RREQ, the destination D updates its routing table by adding a reverse route to the source S, and sends a *Route Reply* (RREP) message in unicast

to the next hop towards S. The RREP can be optionally sent by an intermediate node if it has a fresh route to the requested destination. Intermediate nodes receiving a RREP add a route to D in their routing tables, and forward the updated message to the next hop towards S. Once receiving the RREP, a bidirectional link is established between S and D. If multiple RREP messages are received by S, the route with the shortest hop count is selected.
Route maintenance: to maintain the discovered routes, a route maintenance phase is performed using two additional messages. First, the Hello message, which is sent periodically among neighbor nodes to inform each other about their connectivity. Receiving a Hello message from a node proves that there is an active route through that node. Second, the *Route Error* (RERR) message is sent when a node detects a link break in an active route towards nodes using that link to send their packets. A last important feature of AODV is the use of sequence numbers in packet header to ensure the consistence of route information and prevent routing loops.

2.2.3.3 The OLSR routing procotol

OLSR is a proactive protocol developed by *Institut National de Rercherche en Informatique et en Automatique* (INRIA) [75]. The key concept of this protocol is the use of *Multi-Point Relays* (MPRs) to forward broadcast messages during the routing flooding process.
To build a vision of the network topology, each node performs several steps to obtain and maintain different routing information. First, a neighbor discovery process is initiated using Hello messages dissemination in order to perform a link sensing. Then, each node has the list of its 1-hop neighbors and selects among them a set of MPR nodes, and then reach all its 2-hops neighbors. Second, each node declares its MPR selectors by broadcasting *Topology Control* (TC) messages. The topology information received in TC messages are saved in the topology table, and later used to know the shortest routes to other network nodes.
The OLSR routing scheme ensures an optimized mechanism that effectively reduces the traffic generated by broadcast control messages. Moreover, improving the routing information dissemination still provide optimal routes in terms of hop count and reduces the number of control messages. OLSR is particularly suited for large and dense networks as the technique of MPR works well in this context.

2.2.4 Discussion

Many research works proved that AODV ensures much better performance than other ad hoc routing protocols [76], [77], [78]. On the other hand, many research works presented a high importance of AODV in many smart deployments and real time applications such as, *Voice over IP* (VoIP) [79] and environment monitoring systems [80].
Providing a secure routing is challenging in MANETs; network entities are in charge of providing routing services cooperatively. Moreover, most of ad hoc routing pro-

tocols do not have an adequate security control to prevent misbehaving nodes from participating in the routing process. Consequently, the routing function is targeted by a large number of security attacks, ranging from passive eavesdropping to active denial of service. This latter, is an outcome of one or more malicious behaviors known as availability attacks. The network availability is a crucial security attribute, without which the network is considered as non-functional.
In this thesis, we are particularly interested by security attacks threatening the availability of MANETs' routing layer. In the next section, we reveal security vulnerabilities that can deny the ad hoc routing services, and provide a novel taxonomy of DoS attacks before presenting security solutions in the following chapters. In the rest of the thesis, we use AODV as the base routing protocol, due to its performance superiority over the other protocols.

2.3 Routing availability challenges: DoS attacks

According to the *Internet Engineering Task Force* (IETF), the *Request For Comments* (RFC) 4949 [81] defines an attack as "an intentional act by which an entity attempts to avoid security services and violate the security policy of a system". In addition, the combination of several attacks can lead to more potential attacks which are difficult to detect. A DoS attack is an attempt to make a machine or network resource *unavailable* to its intended users. In the context of MANETs, DoS attacks consume not only the scarce system resources, such as bandwidth, battery energy or *Central Processing Unit* (CPU) cycles, but also keep legitimate users away from the network [82]. Next, we discuss the security challenges in MANETs, especially those threatening the availability of routing services.

2.3.1 Attributes of Denial of Service

Ad hoc routing services may be threatened by different types of DoS attacks. An attacker may consume the resources of a legitimate node by sending it an excessive traffic, or isolate it from the routing process. DoS attacks may have different levels of severity, where the highest level consists of a disabling of the entire network.
To address this problem, it is mandatory to understand the techniques, objectives and damages of DoS attacks. Wood and Stankovic [83] characterized these attacks using five main attributes, which can help to find ways to mitigate attacks by prevention, detection and recovery:

- **Attacker:** it may be classified as either internal if it is part of the network, or external if it has no knowledge of the network. Depending on the attacker's goal, it can have multiple behaviors ranging from a passerby to a terrorist.

- **Capability:** knowing what an attacker is capable of is important for defending the network. The capability of a DoS attack can be described by the number of attackers, their coordination, technical capabilities and area of influence.

- **Target:** the type of a service and its importance are factors that affect the overall risk and constrain solutions. The loss of key services such as routing or directory services may disrupt the entire operation of the network.
- **Vulnerability:** designates the weaknesses in the network, through which an attacker may gain unduly exercise privilege. DoS attacks can be perpetrated by exploiting low-tech physical or logical flaws.
- **Result:** DoS attacks may have different impact levels on the network depending on the intention of the attacker. Under such attacks, the targeted service may be troubled, disrupted, degraded or totally disabled.

This taxonomy helps to identify the profile of an adversary, and then customize the appropriate security countermeasure. However, this abstract description is not sufficient to formally define the DoS. In other words, it is crucial to understand how and why the service can be denied, not only based on the profile of the attackers, but also on the possible scenarios that can lead to such a situation.

2.3.2 Taxonomy of DoS attacks in MANETs

Most research works used the DoS term to define some class of security threats against some network services, or alternatively with flooding attacks in other security fields such as cloud computing [84]. According to [85], DoS attacks are known as *availability* attacks, namely, security attacks that threaten the availability of routing services. In the context of ad hoc routing, we define DoS as "a series of elementary malicious activities that can reduce or completely deny the routing services". In other words, DoS itself is not an attack, but an eventual outcome of a sequence of many attacks.

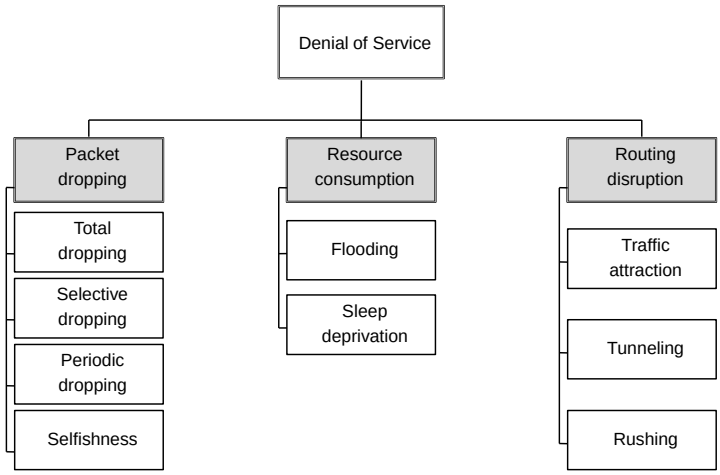


Figure 2.5: Taxonomy of DoS attacks against routing services in MANETs.

Past research works studying DoS attacks in MANETs did not adequately describe the techniques used to launch these attacks. In [86], a few DoS attack scenarios at the network layer were studied. However, there is no clear classification of possible DoS attacks in MANETs and their impact on routing services.
The authors of [87] attempted to provide several system requirements to detect DoS attacks in MANETs. However, they considered DoS as a single attack that overwhelms the network and deprives legitimate users from using network resources. Both of these past works did not really present an in-depth security analysis of DoS attacks, in other words, they did not present a clear taxonomy of attack scenarios leading to a potential denial of routing services.
In this section, we intend to propose a complete classification of threats against the availability of routing services. Figure 2.5 depicts our own taxonomy of DoS attacks that has a tree structure as proposed in [88]. In an *attack tree structure*, the root represents the goal of the attack; in our case, the goal is the DoS. An intermediate node of the tree is a more specific goal (i.e., subgoal) and a leaf is an attack mechanism. Preparative attacks such as, message interception and message forging are omitted from this taxonomy and are out of the scope of our security analysis.

Table 2.1: DoS attacks simulation parameters.

Parameter	**Value**
Coverage area	800 m × 800 m
Transmission range	250 m
Simulation time	300 sec
Mobility model	Random Waypoint
Antenna	OmniAntenna
Minimal/ Maximal speed	[5 − 20] m/s
Routing protocol	AODV
Traffic type	UDP − CBR
Packet size	512 bytes
MAC layer type	IEEE 802.11p

For each attack subgoal of the proposed taxonomy, we describe the protocol vulnerabilities that can be exploited by the malicious node and the possible mechanisms leading to that subgoal. Besides of this description, we present a performance evaluation of MANETs under some attack scenarios that we simulated using *Network Simulator 2* (NS2). In table 2.1, we provide the common experimental parameters of simulations studied throughout this chapter.

2.3.3 Packet dropping attacks

In MANETs, nodes trustworthiness assumption makes the network vulnerable to many security attacks that can disrupt or disable routing services. Packet drop-

ping attacks were widely studied due to their severe impact, especially on packet forwarding. Generally, packet dropping may occur when a *selfish* node does not forward packets in order to save its energy, or when a *malicious* node drops some or all packets it receives [89]. Figure 2.6 illustrates how a malicious node intercepts the traffic before performing a packet dropping attack.

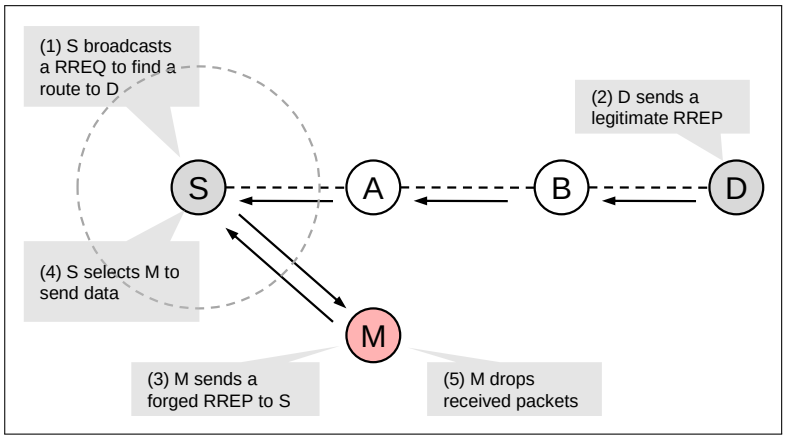


Figure 2.6: Packet dropping attacks in MANETs.

A Blackhole attack occurs when a malicious node advertises itself as having the shortest route to the destination by providing a high sequence number or a low hop count in packets it forwards to other nodes. Once the malicious node receives packets, it absorbs them instead of forwarding them to their destination.
Grayhole attack is a another type of dropping attacks in which a malicious node selectively drops the received packets, in other words, drops some packet types and forwards other ones. For instance, a Grayhole node having the intention to disrupt the route establishment may drop RREP packets and forward other packets.
Similarly, a malicious node may perform a periodic dropping by changing its behaviors over time. In other words, it drops packets sometimes and forwards them normally other times in order to still in the network as long as possible without being detected [90].
Routing services may also be disrupted by selfish nodes which try to save their resources by not cooperating with others in route establishment and packet forwarding. According to [91], a selfish behavior may use one of the following models:

1. *Selfish forwarding model* represents the nodes that refuse to forward data packets while still participate in the route discovery phase.

2. *Selfish routing model* represents the behavior of the nodes that participate neither in route discovery nor in data forwarding phase.

3. *Energy-driven selfish behavior model* combines the two models in order to provide a psychological explication to the selfish behaviors.

Selfish behaviors are not really attacks; their aim is to minimize the chances of being included in routes for which it is neither source nor destination.

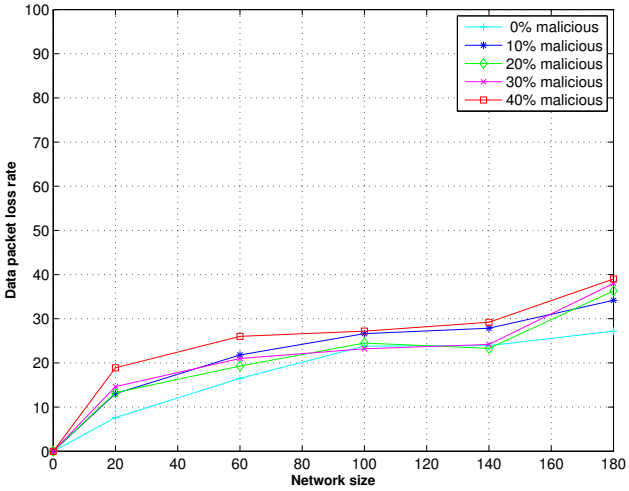


Figure 2.7: Data packet loss caused by dropping attacks.

Packet dropping attacks were largely studied due to their potential damage they cause for ad hoc routing services. These attacks significantly decrease the network performance; they may lead to a high packet loss ratio and low packet delivery ratio. Moreover, the packet loss caused by dropping attacks is usually confused with those caused by packet collusions, especially if the attacker performs a periodic dropping. To show the impact of packet dropping attacks on routing services, we performed many simulation scenarios using NS2. The network size is changed from 20 to 180 nodes, and the percentage of malicious nodes performing packet dropping attacks is between 10% and 50%.
Figure 2.7 shows the data packet loss caused by these attacks in function of network scale and the percentage of malicious nodes. We note that more the monitoring increases in the network the more the data packets are lost. On the other hand, we note a significant percentage of data packet loss in the case of large networks. When the percentage of malicious nodes is equal to 20%, 35% of packets are lost in the case of network size equal to 180 nodes.

2.3.4 Resource consumption attacks

In most reactive routing protocols, routing information dissemination and route discovery phase are performed using a flooding-based scheme. In the case of AODV, a RREQ message is diffused once a sender has data to send and there is no fresh route towards the intended receiver. During the route discovery process, the sender should satisfy the following conditions:

- The generation rate of RREQ messages should be less than a *RREQ Rate Limit*.

- After broadcasting a RREQ, the sender should wait for a RREP. If the latter is not received within a *Network Traversal Time*, the node may perform another route discovery until it reaches a maximum number of *RREQ Retry Times*.

- Time intervals between repeated route request trials should satisfy a *Binary Exponential Back-off* to prolong the waiting times for the next new transmission of RREQ packets.

However, the flooding may have a malicious intention when an attacker broadcasts a large number of useless packets to exhaust the communication bandwidth and degrade the routing services. A malicious node may perform a *RREQ flooding* attack by sending an excessive number of RREQ packets towards out-of-domain IP addresses in order to cause a DoS for one or more network nodes.
A malicious node would also perform RREQ packets retries without waiting for the arrival of RREP packets or a large number of RREQ packets with the maximum *Time-To-Live* (TTL) value in a burst manner. Since the destination IP addresses are invalid, no node could answer RREQ packets, and the reverse routes will be stored for a longer time in network nodes' routing tables. Consequently, the whole network may be crowded with fraudulent RREQ packets, which can disrupt the forwarding services and exhaust the resources of network nodes.

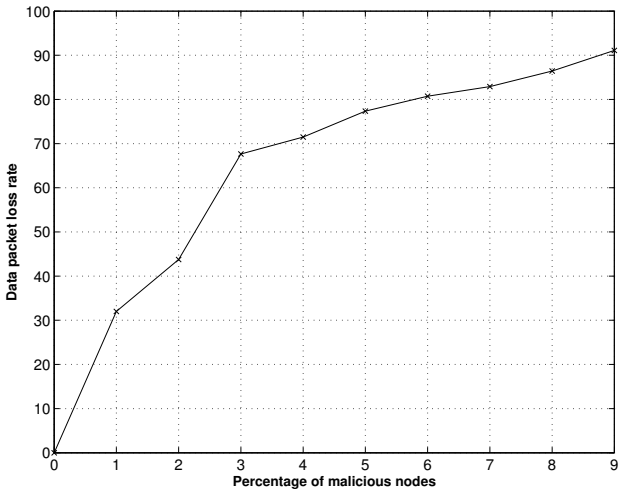


Figure 2.8: Data packet loss caused by RREQ Flooding attack.

To show the impact of this class of attacks on routing services, we performed a number of simulations using NS2 according to the experimental parameters showed in table 2.1.
Five communication sessions are established between five pairs of nodes. Flooding attack is launched by several nodes in the network, which generate an excessive number of malformed RREQ messages. Figure 2.8 depicts the data packet loss ratio caused by RREQ flooding attack during the simulation period.
Data flooding attack occurs after a route is established between the attacker and one

or more legitimate network nodes. After setting up a route to a legitimate node, the attacker forwards a high number of useless data packets along the path to disrupt the normal packet processing of targeted nodes [92], in order to exhaust its battery power and then isolates it from the network.
Sleep deprivation attack is another method which can be performed by an attacker to disable the routing services provided at a legitimate node. The attacker launches a sleep deprivation attack by interacting with the targeted node in a manner that it appears as legitimate; however, the purpose of these interactions is to keep the attacked node out of its power conserving sleep mode [93].

2.3.5 Routing disruption attacks

Ad hoc routing is a service provided cooperatively by network nodes without any administration or central supervision. In terms of security, the number of vulnerable points is proportional to network size. Disrupting one node may cause the disruption of the entire network. In this context, we are interested in attacks which disrupt the normal execution of routing by diverting the packets from their normal direction.
A malicious node may attract the traffic by advertising itself as a best possible node having a route towards some destination in order to deceive other nodes and force them to use that route more frequently for packet forwarding. Such misbehavior is known as *Sinkhole Attack*, which can be established by a malicious insider or a resourceful outsider. For instance, in the case of AODV routing protocol, an attacker can modify or create a RREP message that announces a sequence number larger than that in a received RREQ. Therefore, the fresh route provided by the malicious node guarantees that other nodes will select it as a next hop to forward packets towards the requested destination. Figure 2.9 illustrates how a Sinkhole node M attracts all nodes that want to send data to a destination node D.

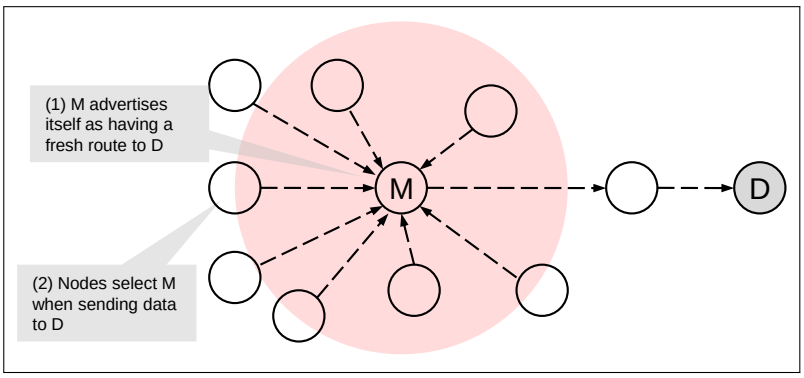


Figure 2.9: Sinkhole attack in MANETs.

Similarly, a malicious node can dominate the route discoveries launched by a legitimate node by forcing the routing protocol to malfunction. Being a flooding-based

routing protocol, AODV limits the number of broadcast RREQ packets using a *Back-off waiting period*, which may be exploited by an attacker to perform a *Rushing attack* [94]. In this case, if the RREQs sent by a malicious node reach the neighbors of a requested destination first, then any route discovered towards this destination will include that malicious node.

To attract the nodes and disrupt the route discovery process, two colluding nodes may also conduct a *tunneling attack*, which is widely known as *Wormhole attack*. This attack is feasible even when the network provides confidentiality and authenticity without requiring any cryptographic knowledge. Analytic and simulation results obtained in [95] demonstrate that a strategic placement of Wormhole nodes can disrupt and control an average of 32% of all communications across the network. If the tunneling is performed without any malicious intention, the attacker actually ensures more efficient connections in the network. However, the powerful position of a Wormhole node can be exploited in a variety of ways, such as packet dropping or data alteration.

The traditional method of packet tunneling consists of a wired or a long-range directed wireless link between two colluding nodes. In a typical Wormhole attack, two colluding nodes make an invalid link between them called a *tunnel*. The packets received by the first malicious node are forwarded through the tunnel, and relayed to another attacker located in another point in the network.
Furthermore, by simply switching the Wormhole link on and off, the attacker can trigger a route oscillation within the network, thus leading to a DoS attack [45].
A more sophisticated method called *packet encapsulation* is used to create a tunnel between malicious nodes. In this type of Wormhole, a malicious node located near to a source of traffic encapsulates the received packets and forwards them to another colluding node near to the requested destination. In its turn, the second malicious node decapsulates the packets and forwards them in such a way that the actual hop count does not increase during the traversal.

In the example illustrated in figure 2.10, when node X hears a RREQ coming from node S, it transmits this RREQ to another colluding node Y at a distant location near the destination. Then Y rebroadcasts the RREQ. The neighbors of Y receive the RREQ and drop any further legitimate RREQs that are coming from legitimate multi-hop paths. As a result, the route between the source and the destination includes the malicious nodes performing the Wormhole attack [96]. This prevents nodes from discovering legitimate paths that are more than two hops away.
To show the impact of these attacks on ad hoc routing services, we simulated the Wormhole attack that use the packet encapsulation technique. In addition, the malicious nodes performing the Wormhole attack drop the packets they intercept instead of forwarding them. The simulated network is composed of 30 nodes, where the number of malicious nodes ranges from 1 to 12 nodes. The rest of configuration parameters are showed in table 2.1.

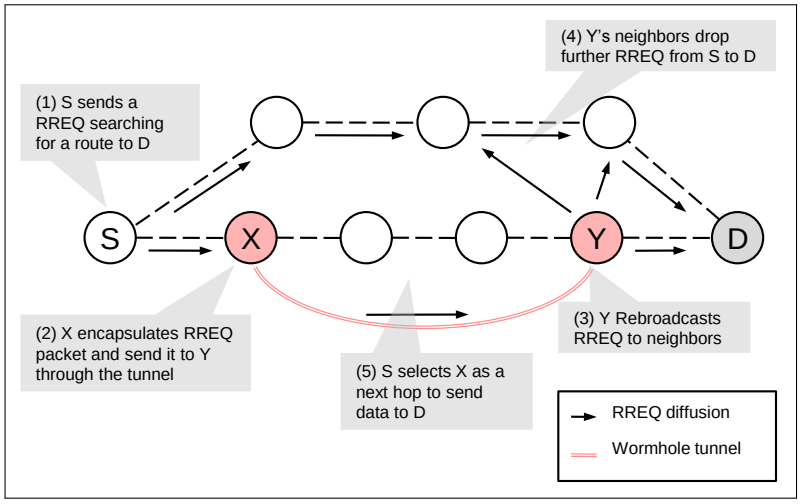


Figure 2.10: Wormhole attack in MANETs.

Figure 2.11 depicts the data packet loss caused by Wormhole attacks in MANETs. We note that more than 80% of packets are lost when 3 pairs of nodes perform the Wormhole attack scenario that we already described.

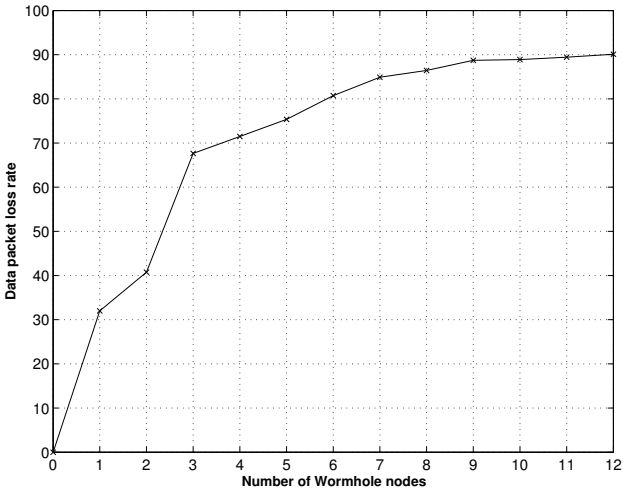


Figure 2.11: Data packet loss caused by Wormhole attacks.

Another type of routing disruption may occur when a malicious node convinces two distant nodes that are neighbors using packet relaying or a *Hello flooding* attack. The former misbehavior can be performed by one or more Wormhole nodes relaying packets between legitimate nodes. In the latter case, a malicious node exploits the route maintenance phase of a reactive routing protocol by generating Hello packets with a high transmission power and without respecting the *Hello Interval* [97]. When the targeted node receives such Hello messages it consider that the malicious

node belongs to its transmission range and may be used to forward packets.

2.3.6 DoS attacks versus routing vulnerabilities

DoS attacks are the most crucial challenge against routing services availability in MANETs. Ensuring privacy and authentication in MANETs does not imply that the network is totally protected. Many security vulnerabilities can be exploited in routing protocols to launch an attack and disrupt the network availability.

DoS attack	Misbehavior	Attack type	Vulnerability	Impact
Packet Dropping	Total dropping	Blackhole	SN or HC	Data Packet Loss
	Selective dropping	Grayhole		
	Periodic dropping	Jellyfish dropping		
	Selfishness	N/A		
Routing disruption	Attraction	Hello flooding	Hello interval time	Malicious Traffic Interception
		Sinkhole	HC	
	Tunneling	Wormhole		
	Protocol distortion	Rushing	Back-off period	Forwarding Invasion
Resource consumption	Flooding attacks	RREQ flooding	RREQ rate limit	Routing Table Overload
		Data flooding	N/A	Network Throughput Damage

Table 2.2: DoS attacks against ad hoc routing services (case of AODV).

For instance, a malicious node may falsify the hop count of a routing message in order to hijack or redirect the traffic to another colluding node, or convince legitimate nodes to use Wormhole links to forward packets. On the other hand, the sequence number of a route can be altered by a malicious node in order to prevent legitimate nodes from using optimal routes.
In table 2.2, we summarize the most potential threats that can disrupt the availability of routing services and lead to a DoS attack based on the taxonomy presented in 2.3.2. For each attack mechanism, we show the vulnerabilities it exploits to take place and its impact on routing performance.

2.4 Conclusion

In this chapter, we introduced the importance of mobile ad hoc networks and their different applications, especially in Smart city deployments. Then, we presented the fundamental concepts of routing in MANETs and explained the routing process of two well-known routing protocols OLSR and AODV.
The second section was dedicated to challenges of routing services availability in MANETs. Then, a novel definition of DoS attacks was presented by describing their attributes, and classifying their mechanisms using an attack tree. For each attack mechanisms at the lowest level of the tree we described the vulnerabilities that may be exploited by attackers, and we proved through experimental results the

impact of these attacks on routing performance.
In the proposed attacks taxonomy, we omitted preparative misbehaviors used in some attacks mechanisms such as, message interception, forging, etc. We focused on well-known attacks that we considered as potential threats against the availability of routing services in MANETs. In the next chapter, we address the problem of DoS attacks at the routing layer in MANETs by discussing some research works that were proposed in the literature.

Chapter 3

Security mechanisms against DoS attacks in MANETs

"Each problem that I solved became a rule, which served afterwards to solve other problems."

– *Rene Descartes*

Contents

3.1 Introduction

The lack of a central administration, and the high dependence on inherent node cooperation make ad hoc routing services vulnerable to many threats. To address the vulnerabilities discussed in the previous chapter, many security mechanisms dedicated for ad hoc routing layer were proposed in the literature.
In this chapter, we present a state of the art of ad hoc routing security mechanisms within the strategies mentioned above. For each DoS attack technique described in figure 2.5, we detail some representative solutions, and we discuss their advantages and limitations on network performance. The discussed research works were selected as a trade-off between the most cited and the most recent works, to provide a rich and comprehensible state of the art.
Among DoS attacks, we focus on packet dropping attacks and their countermeasures in the aim of clarifying our contributions in comparison with existing solutions.

3.2 Mechanisms against packet dropping

Ad hoc routing services rely on the cooperation of all network nodes. Thus, selfishness is one of the major security problems in ad hoc networks. A selfish node aims to preserve its own resources while using the services of others and consuming their resources. As we have already mentioned in 2.3.3, with Blackhole and Grayhole attacks, a node malicious node drops the totality or a part of packets it receives instead of forwarding them.
To highlight potential research works proposed to secure MANETs against this class of attacks, we detail in the following some security solutions using three classes. First, we present solutions that use cryptographic tools to prevent packet dropping attacks. Second, we describe those based on *trust management* to detect selfish and malicious nodes from routing operations. Finally, we discuss some *classification frameworks* that were proposed to detect packet dropping attacks in MANETs.

3.2.1 Cryptography-based mechanisms

Most of these mechanisms were proposed to allow the protocols to authenticate network nodes and guarantee the integrity of routing information. Most of these solutions were proposed as preventive strategies against the malicious modification of routing information, which occurs usually prior to launching a packet dropping attack.

Authenticated Routing for Ad hoc Networks (ARAN)

This security protocol was proposed in [18] to ensure routing messages' integrity and non-repudiation based on public key cryptography. ARAN requires the use of a trusted *Certificate Authority* (CA) server, whose public key PK_{CA} is known to all valid nodes. Each node denoted by A has to obtain a certificate $cert_A$ before

entering the network, where $cert_A =< IP_A, PK_A, t, e >_{SK_{CA}}$. The obtained CA belongs to the node having the IP address IP_A, created at a time equal to t, expires at a time equal to e and having SK_{CA} as a private key.
Each node running ARAN uses a secure packet signed with its private key called *Route Discovery Packets* (RDP). At the beginning of a route request initiated by a node S towards a node D, a RDP message is constructed and diffused to neighbor nodes, containing the IP address of D, the certificate of S, a monotonically increasing SN and a timestamp t, RDP$=< IP_D, cert_S, SN, t >$. The freshness of RDP message can be verified by intermediate nodes using the IP_S found in $cert_S$ in conjunction with SN.
After receiving a RDP message, each intermediate node I verifies the signature of the node from which it received that packet, removes it from the message, then adds its own signature and certificate to the message before rebroadcasting it: $<<< IP_D, cert_S, SN, t > SK_S > SK_N, cert_N >$. Once D receives a RDP message, it considers only the first received one and ignores more further messages. Then, a reply process is performed using a *Reply packet* (REP) that is sent using the reverse path traversed by the RDP message.
ARAN is one of the most known secure routing protocols, which provides authentication, non-repudiation and integrity of routing information, at the price of weighty cryptographic operations and the use of CA server.

Secure AODV (SAODV)

Zapata *et al.* [17] proposed this secure routing protocol to protect the AODV routing information. SAODV needs a CA server to manage *Public Key Infrastructure* (PKI) in order to sign routing messages sent by network nodes. A hash chain is used to authenticate the only mutable field in AODV messages, the *Hop Count* (HC) field. According to SAODV, when a node wants to establish a communication session, it has to generate a random seed s and a hash function $h^{Max\_HC}(s)$, and puts them into each RREQ.
When an intermediate node receives the RREQ, it increases the HC field, and replaces s by its own hash s'. The integrity of HC field is verified at any node by checking whether $h^{Max\_HC}(s)$ is equal to $h^{Max\_HC}(s')$.
Another aspect of SAODV is that it adds some measures to prevent malicious nodes which try to attract traffic by deliberately increasing the SN field. Although that SAODV guarantees the integrity and authentication of routing information, it cannot totally prevent attacks on HC using hash chains. Moreover, nodes may spend a long time in computing signatures, and become overloaded, which degrades the throughput and network delay.

Hierarchical Secure Routing against Blackhole attacks (HSRBH)

This secure routing protocol was proposed in [20] to discover safe routes against Blackhole attacks using symmetric key cryptography. The core idea of this protocol

is to divide the network into various groups organized in a tree structure with the root of the tree called *Group Leader*.
An intra-group key among neighbors of the group leader and an inter-group key between the two neighboring group leaders are established. To detect Blackhole attacks, the source node randomly sends a control message to the destination and waits for receiving an acknowledgement.
The acknowledgement message sent by the destination includes the *Message Authentication Code* (MAC) which is created using the shared key between the source and the destination to verify whether the route is secure. If the verification is not successful, then the route is considered as infected by a Blackhole node.

Triangular-based Encryption (TE)

This mechanism was proposed in [21] to prevent Blackhole nodes from participating in the AODV route discovery phase. A node requesting a route towards a destination must include a *clear text* in the RREQ before broadcasting it. When the destination receives the RREQ, it encrypts the plain text and sends a RREP message to the source containing the *cipher text*. As the RREP packet contains the cipher text, the destination node must be reached.
The main assumption used by the authors to detect Blackhole nodes is the fact that and attacker sends RREP messages without consulting the routing table. Therefore, it is not possible that it can obtain the cipher text while creating the RREP packet. Consequently, the presence of Blackhole nodes along the route can be verified by the source node by checking whether the packet has the matching cipher.
The authors argue that their proposed method guarantees that the RREP messages are forwarded only using legitimate nodes. However, they have not presented any details about the encryption algorithm used to secure the route establishment.

Adaptive Encryption (AE)

Nekkanti and Lee [19] proposed an extension for the AODV routing protocol using a *trust factor* and a *security level* at each node. The trust factor and the level of security assigned to the information flow decide what level of encryption is applied to the current routing information at a source or an intermediate node.
The core idea of this approach is to ensure an *adaptive encryption*, by masking the routing information only from the non-trusted nodes instead of masking it from all the nodes, in order to save both time and energy. Therefore, when a node S wants to find out a route towards a destination D, it sends the RREQ messages according to the following formula:

$$S \rightarrow broadcast : \{RREQ, SN, P_bD[S_{ID}], D_{ID}, SL\}$$

where $P_bD[S_{ID}]$ is the encrypted source ID with the destination's D public key, D_{ID} is the destination ID and SL is the security level set by the application.
When an intermediate node B receives the RREQ, it looks up its trust table for

each of its neighboring nodes and encrypts its own information with its private key, appends it to the source information and then encrypts the whole information with the public key of the destination node:

$$B \rightarrow broadcast : \{RREQ, SN, P_bD[P_vB[B_{ID}]], P_bD[S_{ID}], D_{ID}, SL\}$$

where $P_bD[P_vB[B_{ID}]]$ is the encrypted intermediate nodes' $ID(B)$.
When D receives the RREQ, it verifies if the path includes any bad nodes using the intermediate node list. If a bad node is found, the RREQ is discarded. Otherwise, D generates a flow-ID and encodes it with the public keys of intermediate nodes in the order they would receive, and then it broadcasts the RREP to its neighbors. If B and C are the intermediate nodes, D sends the following RREP:

$$D \rightarrow broadcast : \{RREP, P_bC[F_{ID}, P_bB[F_{ID}, PbS[P_vD[F_{ID}]]]]\}$$

Finally, when S receives the RREP, it first applies its private key and then the public key of D, and then gets the flow-ID generated by D, which completes the route discovery process. If B is the intermediate node near to S, the packets are sent to D using the same flow-ID of the received RREP: $S \rightarrow B : \{F_{ID}, Data\}$.

Secure Enhanced AODV (SEAODV)

Li *et al.* [4] proposed a secure protocol that employs a *Blom's key pre-distribution* scheme [98] in conjunction with the *enhanced HELLO message* to establish *Pairwise Transient Key* (PTK) and then distributes a *Group Transient Key* (GTK). PTK and GTK provide a hop-by-hop authentication routing solution, in which routing messages are protected at every hop during the route setup process.
To secure the route discovery process, a MAC field is appended to the RREQ message before diffusing it to 1-hop neighbors. The MAC is obtained using the GTK of the node generating the route request A and the rest of message fields M: $A \rightarrow * : RREQ : [M, MAC(GTK, M)]$.
Upon receiving the broadcast RREQ, each neighbor compares its GTK with that of A. If there is a match, it computes the corresponding MAC with the received message and the GTK before updating its routing table and setting up the reverse route back to A. When RREQ reaches a node having the right to reply according to AODV specification, it computes a new $MAC(PTK, M)$ using the PTK of the next hop in the reverse route towards A. If a node B needs to unicast the RREP to the node A it sends the following message: $B \rightarrow * : RREP : [M, MAC(PTK_{BA}, M)]$.
When A receives the RREP from B, it verifies whether PTK_{BA} is in its PTK group, and then updates the HC field in the RREP and its own routing table before setting up the forwarding path towards the destination.

Secure records of packet delivery information

This work was proposed in [5] to solve the problem of Blackhole attacks by securing the history records of packet delivery information at each node encountered along

the path from a source to a destination. The authors used public key cryptography to encrypt the routing information. Each node saves the history of records created by encountered nodes, and keeps in its memory the packets' receiving and forwarding records created by the encountered nodes. There are two tables generated at each node for storing these records. *Receiving Record Table* (RRT) is used by a node to keep packet exchange records generated by its encountering nodes and a *Self Record Table* (SRT) which maintains the records it generates for each node encounter.
When two nodes interact, they validate the history records from each other, and determine the legitimacy of the encountered nodes, and further detect the presence of a Blackhole attack. During the exchange of history information, each node creates a record that contains its ID and that of the encountered node, the number of received and forwarded messages "from" and "to" this node respectively, and the current timestamp t. Then, the node signs the record using its private key. The stored information is used to verify if an encountered node has normally forwarded the received packets during a history window interval.
Although this mechanism shows a good detection capability of Blackhole attacks, the authors have not provided details about the exchange period between nodes. Therefore, if the history size requires a transmission time greater than the exchange period, then the collected information is unlikely to be sufficient to classify the encountered node.

3.2.2 Trust management

In the context of network security, a trust management is a risk management with a particular emphasis on the authentication of entities under uncertainty and decision making based on cooperation with unknown entities [99]. Trust management was employed in MANETs to establish a so-called *reputation system* to satisfy the following security goals:

- Malicious nodes detection: the trust value assigned to a node is obtained based on its behavior. A node is considered as legitimate if it has a high trust value (i.e., greater than certain threshold), while a node having a low trust value is considered as malicious [100].
- Malicious nodes isolation: in the case of trust-based routing, packet forwarding is performed among nodes having a high trust value. Therefore, a node having a low trust value is progressively isolated from the routing process [101].
- Cooperation reinforcement: malicious nodes are susceptible to be punished by network nodes, which can motivate them to cooperate in the routing process [102].

Reputation systems were used in a variety of applications, among them are the selection of good peers in a P2P network, the choice of transaction partners for online auctioning and misbehaving nodes detection in wireless networks [103].
In case of a MANET, the *reputation* of a node refers to how good the node is in terms

of its contribution to routing activities in the network. Most of reputation systems use the monitoring mechanism called *watchdog*, in order to collect information of routing behaviors.
In the following, we discuss some research works that have used trust management to establish a reputation system to detect, isolate and reinforce the malicious nodes to cooperate in packet routing process.

Watchdog and Path Rater

The concept of watchdog was introduced by Marti *et al.* [22] to mitigate the effects of routing misbehavior in the *Dynamic Source Routing* protocol (DSR) [104]. In this approach, the authors assume that wireless interfaces support *promiscuous mode* operation, which means that if node A is within the transmission range of a node B, it can overhear communications "to" and "from" B even if those communications do not directly involve A. Each node supports two components: *watchdog* and *path rater*.
The watchdog maintains a buffer of recently sent packets and compares each overheard packet with the packet in the buffer to see if there is a match. If so, the behavior is considered as normal, and the packet is removed from the buffer. If a packet remains in the buffer for longer than a certain period, the watchdog keeps track of that node and declares it as misbehaving if the period exceeds a certain threshold.
Each node maintains a rating for every other node it knows about in the network. The path metric is calculated by averaging the node ratings in the path. Based on the knowledge of nodes' rankings, the path rater can choose the node that is most likely to deliver the packets.
The main drawback of this approach is the fact that the reputation considers only the direct observations. Many research works have introduced a second hand information, based on the recommendations of other network nodes when calculating the global reputation value such as, CORE [23] and CONFIDANT [24].

Cooperative Reputation (CORE)

This work was proposed in [23] to enforce nodes cooperation in MANETs. The global reputation of a node is evaluated by combining three types of reputations:

- **Subjective reputation**: is evaluated only by considering the direct interaction between a subject and its neighbors. A subjective reputation at time t from subject s_i point of view is calculated using a weighted mean of the observations' rating factors, by giving more relevance to the past observations.

- **Indirect reputation**: adds the possibility to use information provided by other members of the community to calculate the final value to the reputation of a subject. This type is considered as a second-hand reputation, and is used optionally in CORE.

- **Functional reputation**: adds the possibility to calculate a global value of a subject's reputation that takes into consideration different functions as evaluation criteria such as, packet forwarding ($f = PF$) or routing function ($f = R$).

The monitoring process is performed by storing several essential information of packets passed by the node. Each packet is identified by a *unique identifier*, the addresses of its sender S and its receiver D, and a hash value of the *data payload*: $< UID, IP_S, IP_D, h(payload) >$.
CORE compares each overheard packet to what it is excepting. In case of data alteration or packet dropping, the reputation of a node decreases, otherwise it increases. When a node A wants to calculate a reputation on a node B at a time t, it combines different reputation evaluations as follows:

$$r_A^t(B) = \sum_{f \in \{PF,R\}} w_f \{r_A^t(B|f)\} + \sum_{z \in N_A} \lambda_z \{r_z^t(B|f)\} \tag{3.1}$$

where w_f denotes the weight of the function f, and N_A regroups the direct neighbors of node A. If the global reputation without second-hand reputations, then $r_z^t(B|f)$ is set to 0. Otherwise, $\lambda_z = r_A^t(z|f)$ denotes the weight on the indirect reputation $r_z^t(B|f)$.

Cooperation Of Nodes Fairness in Dynamic Ad hoc NeTworks (CONFIDANT)

This work was proposed in [24] in the aim of establishing a consistent reputation system based on both direct and indirect observations, in order to detect and isolate misbehaving nodes in MANETs.
CONFIDANT consists of four components: the *monitor*, the *trust manager*, the *reputation system* and the *path manager*. Unlike CORE, with CONFIDANT a node monitors all its neighbors and locally looks for deviating neighbors. The latter is ensured by either listening promiscuously to the transmission of the next node or by observing route protocol behavior.
The trust manager of a node deals with *incoming* and *outgoing* ALARMS to warn others of malicious nodes. *Outgoing* ALARM messages are generated by the node itself after having observed a malicious behavior.
The recipients of these messages are called *friends*, which are administered in a *friend list*. The trust manager maintains an *alarm table* containing information about received alarms, and a *trust table* to manage the trust level of incoming ALARM messages. Trust management is performed using a method similar to that proposed in *Pretty Good Privacy* (PGP) [105], which defines four levels of trust: *friend*, *marginal*, *unknown* and *enemy*.
The reputation system component manages a table consisting of entries for nodes and their rating. The rating is changed only when there is a sufficient evidence of malicious behavior that is significant for a node and that has occurred a number of times exceeding a threshold to roll out coincidences.

Finally, the path manager ensures a routing according to the reputation of the nodes in the path, and deletes paths containing malicious nodes.

Trust-based QoS Routing (TQR)

This protocol has some particularity since it takes into account QoS besides the trust management scheme. It was proposed by Wang *et al.* [7] to estimate the available link delay requirement by considering link quality, and incorporating a trust system into the route discovery procedure to enhance the security of the network.
In TQR protocol, each node i derives a trust degree value for each of its 1-hop neighbors. The trust degree value of i in its neighbor j at a time t is defined as the weighted average of two parts. The first one is the direct trust degree $T^d_{i,j}(t)$ of i in j based on i's direct observation of j's packets forwarding behavior at a time t:

$$T^d_{i,j}(t) = \frac{F_{i,j}(T)}{R_{i,j}(T)} \tag{3.2}$$

where $F_{i,j}(T)$ represents the number of packets forwarded correctly by j at time t, and $R_{i,j}(T)$ is the number of packets successfully received by j from i at time t.
The second part is the average of existing indirect trust degrees $T^r_{i,j}(t)$ that the n mutual neighbors of i and j have in node j by recommendation at time t:

$$T^r_{i,j}(t) = \frac{1}{n}\sum_{k=1}^{n} T^d_{k,j}(t) \tag{3.3}$$

The global trust $T_{i,j}(t)$ combining the two parts defined above is obtained as follows:

$$T_{i,j}(t) = \omega_1 T^d_{i,j}(t) + \omega_2 T^r_{i,j}(t) \tag{3.4}$$

where the weight factors ω_1 and $\omega_2 \in [0,1]$ such that $\omega_1 + \omega_2 = 1$. Therefore, when a source node discovers a path to a destination, the trust degree of that path denoted by R_r should be computed according to the trust degree of the l nodes constituting the route r using the following formula:

$$R_r = T_{1,2}(t)T_{2,3}(t)\ldots T_{l-2,l-1}(t) = \prod_{i=1}^{l-2} T_{i,i+1}(t). \tag{3.5}$$

The authors assumed that the initial trust degree value is set to 0.5, which is used to represent a neutral view on unknown nodes. Network nodes are placed in the promiscuous mode; each node can overhear packets forwarded by its neighbors. For instance, $R_{i,j}$ is incremented whenever i finds that j received a packet, and if that packet is forwarded $F_{i,j}$ is incremented. Based on the interactions between i and j, the trust degree is increased after each successful forwarding, and decreased when a failed forwarding is detected.
Besides the trust model, the authors considered the link delay as QoS parameter by

measuring the link quality, and used an *Expected Transmission count* (ETX) as a metric when selecting a route:

$$X_j(t) = \frac{1}{F_j(t) \times R_j(t)} \tag{3.6}$$

where $F_j(t) \times R_j(t)$ is the probability that a probe packet is successfully "sent to" and "acknowledged back" by receiving node. Therefore, $X_j(t)$ denotes the expected number of successful transmissions measured by node j at time t. $\delta_j(t)$ is the total link delay defined to take into account delay determined by the queue buffer size and transmission delay caused by link bandwidth decrease. A new routing metric $C_r(t)$ considering both the trust degree of route and QoS requirements is defined using the following cost metric:

$$C_r(t) = \sum_{j \in r} \delta_j(t) \times (1 - T_{i,j}(t)) \tag{3.7}$$

where r is a sequence of nodes constituting the route between the source and the destination. Finally, using this formula, the authors defined a trade-off between quality and trust degree of the route, which can be adjusted according to an optimization method having a minimum value of $C_r(t)$.

Trusted AODV (TAODV)

Li *et al.* [25] employed this trust model using three modules which implement the following components: basic AODV routing protocol, a trust model based on the subjective logic and the trusted AODV routing protocol. The subjective logic is defined as a logic which operates on subjective beliefs about the world, and uses the term opinion to denote the representation of a subjective belief.
In the proposed trust model, the authors defined an opinion of a node A about the trustworthiness of a node B using a three dimensional metric:

$$\omega_B^A = (b_B^A, d_B^A, u_B^A) \; such \; that \; b_B^A + d_B^A + u_B^A = 1$$

where b, d and u correspond respectively to *belief*, *disbelief* and *uncertainty*. The trust aggregation in TAODV is defined by combining opinions using two possible operations:

- **Discounting combination:** if node A wants to know the trustworthiness of a node C, and node B gives its opinion about C, then A will combine the two opinions: A to B and B to C to obtain a recommendation opinion A to C:

$$\begin{cases} b_C^{A,B} &= b_B^{AB} b_C^B \\ d_C^{A,B} &= b_B^{AB} d_C^B \\ u_C^{A,B} &= d_B^A + u_B^A + b_B^A u_C^B \end{cases}$$

- **Consensus combination:** to get a relative objective evaluation where different nodes have different or contrary opinions about another node:

$$\begin{cases} b_C^{A,B} &= (b_C^A u_C^B + b_C^B u_C^A)/k \\ d_C^{A,B} &= (d_C^A u_C^B + d_C^B u_C^A)/k \\ u_C^{A,B} &= (u_C^B u_C^B)/k \end{cases}$$

where $k = u_C^A + u_C^B - 2u_C^A u_C^B$ such that $k \neq 0$.
Based on the number of successful or failed communications, the node changes its opinions about other nodes using some *trust updating* procedure. To ensure a trusted route discovery in AODV, the trust recommendations are exchanged using three types of messages: *Trust Request* (TREQ) issued by a trust *requester*, *Trust Reply* (TREP) issued by a *recommender* node about a *recommendee* node, and *Trust Warning* message (TWARN) to warn nodes when a malicious behavior is detected. The salient feature of TAODV is that using trust relationships among nodes, there is no need for a node to request and verify certificates all the time. Once the trust relationships are established, the subsequent routing operations can be performed securely based on the trust information.

Friendship AODV (FrAODV)

Another security extension for the AODV routing based on a friendship mechanism was proposed in [26]. FrAODV is based on two algorithms called *RvEvaluate* and *FwEvaluate*, which are used to evaluate respectively the reverse and forward routes, and then build up trusted routes in AODV.
RvEvaluate algorithm is triggered once a node S sends a RREQ to find out a route to a destination D. An intermediate node accepts the RREQ only if the friendship evaluation of the previous hop is positive. In this case, it creates a reverse route and evaluates the friendship of the route denoted by $RvFrRTe$:

$$RvFrRTe = \sum_{i=1}^{h} \frac{PfFrHp_i}{h} \tag{3.8}$$

where $PfFrHp_i$ is the friendship value of the previous hop i, and h is the number of hops from the current node to S. Similarly, when D receives the RREQ, it evaluates the friendship of its previous hop N using the same scheme mentioned above. If the obtained value of $RvFrRTe$ is less than the friendship value of the existing route, D rejects the route. Otherwise, it considers this route as the best reverse route.
In the *FwEvaluate* algorithm, trusted forward routes are built using a similar scheme. This algorithm starts when a destination D sends a RREP as a response to a received RREQ. The RREP is accepted by an intermediate node N, if its previous and next hop are friends. In this case, a forward route is created, and evaluated by node N as follows:

$$FwFrRte = \sum_{i=1}^{h} \frac{FwFrHp_i}{h} \tag{3.9}$$

where $FwFrHp_i$ is the friendship value of the next hop i, and h is the number of hops from the current node to D. When the source S receives the RREP message, it verifies if the next hop is a friend and evaluates the forward route using the same formula mentioned above. If the obtained value is less than the the current routes friendship value, then the new route is rejected. Otherwise, this route is selected as the best friendly forward route.

Fuzzy Petri Net based Trust OLSR (FPNT-OLSR)

FPNT-OLSR is a trust reasoning model based on *fuzzy Petri Net* that was presented in [52] to evaluate trust values of mobile nodes in MANETs. In this model, the trust evaluation is based on four factors: traffic load, packet forwarding rate, average forwarding delay and protocol deviation flag. Based on these factors, a set of weighted fuzzy reasoning rules propositions are developed through a trust evaluation algorithm. The global trust value of a target node is obtained by combining the direct trust of the monitoring node and its neighbors recommendations.
A node supporting FPNT-OLSR runs in a promiscuous mode to intercepts and reads every network packet arriving at the target, in order to collect trust factors. MPR nodes are responsible for monitoring and evaluating their selectors and propagating the obtained evaluation result as recommendations. The trust aggregation process relies on direct trust evaluations and recommendations received from other nodes. Based on trust values of other nodes, each node in the network can calculate a trust-based routing table.
To avoid malicious or compromised nodes, each node maintains *candidate table* TAB_{CAN} to cache all possible path entries towards a destination. The entry having the maximum path trust value in TAB_{CAN} is selected and moved to a trust-based routing table TAB_{PT}.
The authors showed that FPNT-OLSR performs well against many security threats such as, Blackhole and Grayhole attacks. However, the trust computing for every path from the source to a newly added node may result a high overhead and end-to-end delay.

3.2.3 Classification frameworks

Intrusion detection has been frequently used as a second line of defense in MANETs. A simple way to perform intrusion detection is to use a classifier in order to decide whether some observed traffic data is "normal" or "abnormal" [106]. In this section we discuss four classification frameworks that were proposed to detect packet dropping attacks in MANETs.

Finite State Machine (FSM) based mechanism

To distinguish selfish nodes from cooperative ones, Wang *et al.* [27] proposed a method to build up a statistical description node's behavior based on an FSM model of locally observed AODV actions. The proposed technique requires no training data

but instead compares observed behaviors of multiple neighbors against each other, providing a basis for an online local reputation assessment algorithm.
To constitute the information required to detect selfish routing behaviors, a *Local Routing Instance* (LRI) is defined for each node including a subset of transmissions generated by itself and its neighbors. Each LRI is identified by the combination of the source and destination contained in a RREQ message. The FSM is used to describe the behavior of a single node with respect to a single LRI. Each transmission observed by a local node is recorded as a state transition in one or more neighbors' FSM. A sequence of transitions among certain possible states are recorded by a local node according to the observed events performed by a monitored node.
Upon reaching a final state, the FSM is considered complete and the local node stores the completed sequence to derive a matrix T containing the probability of observing each transition state T_{ij}:

$$T_{ij} = 1/N \times \sum_{k=1}^{N} 1(i \to j \in X_k) \tag{3.10}$$

where N is the total number of completed FSMs for the monitored node and 1 is an indicator function.
To detect selfish behaviors, a series of statistical tests are applied to attributes which are extracted from the set of transition matrices for all of the local node's neighbors. Based on these tests, the authors proved how their approach can detect both RREQ and RREP dropping attacks.

Dynamic learning for anomaly detection

Kurosawa *et al.* [28] proposed the use of a dynamic training method, which takes into account the *Destination Sequence number* (Dst_Seq) to detect packet dropping caused by Blackhole attacks. In this solution, it is assumed that the *Dst_Seq* increases largely when a Blackhole attack takes place, and base their detection method on three features:

- x_1: Number of sent RREQ messages.
- x_2: Number of received RREP messages.
- x_3: Average of the differences between the *Dst_Seq* in sent RREQ and those in received RREP messages.

The network state during a time slot i is expressed by three-dimension vector $x_i = (x_{i1}, x_{i2}, x_{i3})$. Then, a mean vector $\bar{x}^D$ using a training data set D of N time slots:

$$\bar{x}^D = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{3.11}$$

Next, the distance from input data sample x to the mean vector $\bar{x}^D$ is calculated using the following formula:

$$d(x) = ||x - \bar{x}^D||^2 \tag{3.12}$$

Finally, the obtained distance is compared to a certain threshold value T_h. If $d(x)$ is greater than T_h, then a deviation from the normal network state is detected, and the state is treated as data including attack. Otherwise, the state is judged as normal, and then the corresponding data set will be used as a training data set. At the end of this procedure, the training data set is updated to be used for the next detection. Then, the mean vector $\bar{x}^D$ obtained from this training data set is used for the detection in the next data set. By repeating this procedure, the authors consider that their proposed scheme is able to detect anomalies in an ad hoc environment.

Dynamic trust prediction

Xi *et al.* [29] proposed a dynamic trust prediction model to evaluate the trustworthiness of nodes, which is based on nodes' historical behaviors, as well as the future behaviors via extended *fuzzy logic* rules prediction. In the proposed model, three types of trust are defined:

- *Node's historical trust*: it is estimated by the packet forwarding ratio composed of two values, *Control packet Forwarding Ratio* (CFR) and *Data packet Forwarding Ratio* (DFR). At a time t, the historical trust value $HTV_{ij}(t)$ denotes the evaluated node v_j's trust value standing on the monitoring node v_i's point of view and calculated using the following formula: $HTV_{ij}(t) = w_1 \times CFR_{ij}(t) + w_2 \times DFR_{ij}(t)$. w_1 and w_2 are the weights assigned to $CFR_{ij}(t)$ and $DFR_{ij}(t)$ respectively, such that $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$.

- *Node's current trust*: considering a monitored node's historical trust and its current capability to provide services, the current trust is computed using a prediction method based on a set of fuzzy logic rules. At a time t, this trust type represents the *trust value* is denoted by $TV(t)$, and $C(t)$ represents the node's capability level on providing packets transmission services at time t. Finally, a specific number of fuzzy sets are defined for $TV(t)$, $TV(t+1)$ and $C(t)$ in order to construct an inference relationship between them.

- *Route trust*: assigned to a route based on the quality of providing services such as packet forwarding. This trust type is denoted by $RouteTV_P$, which is computed according to the intermediate nodes' trust values along an established route between a source and a destination. At a time t, the trust of a route P is equal to the continued product of node trust values in a route P between a source v_s and a destination v_d, and calculated as follows:

$$RouteTV_P(t) = \prod(\{TV_{ij}(t) | v_i, v_j \in P \text{ and } v_i \rightarrow v_j\}) \tag{3.13}$$

in which, $v_i \rightarrow v_j$ means that v_j is the next hop node of v_i. Based on the trust prediction model, a novel on-demand trust based unicast routing protocol for MANETs is presented, termed as *Trust-based Source Routing protocol* (TSR). In this protocol, a source can establish multiple loop-free routes to a destination in one route discovery process, and each route has an evaluation vector composed of hop count and

route trust value. A destination responds with qualified routes as candidates that satisfy the trust requirements of transmitting data packets. The core idea of this protocol is to provide a flexible and feasible approach to choose the shortest route that meets the security requirement of data packets transmission.

Intrusion Detection System (IDS) for dropping attacks

This work was presented in [30] to propose a model of data forwarding to recognize malicious packet dropping behaviors in MANETs. In this solution, the set of successive actions that should be performed by a node N to forward a received packet are:

1. $\overline{Dest}$ event: N is not the final destination.
2. *Rout* event: N has a valid route for relaying packet towards the destination.
3. $\overline{Drop}$ event: N is not a malicious dropper.

A node that wants to forward packets must first try to send a *Request-To-Send* (RTS) packet. The probability that N sends an RTS message denoted by P_{RTS} considers those packets that fulfill the conditions *dest* and *rout*, since these two conditions could be easily determined by inspecting every received packet:

$$P_{RTS} = Pr(RTS|\overline{Dest}, Rout) = (1 - P_{Drop}) \tag{3.14}$$

where P_{Drop} is the probability that the packet is maliciously dropped by N.
Second, N checks if it receives a CTS message. This latter is received from the next hop in the route once the corresponding RTS packet reaches its destination. The probability that a CTS is received given that RTS event is occurred is obtained:

$$P_{CTS} = Pr(CTS|RTS) = 1 - (P_{COL} + P_{MOB}) \tag{3.15}$$

where P_{COL} is the probability for the RTS or CTS packets to be lost due to collisions or channel errors, and P_{MOB} is the probability of packet losses due to broken links caused by mobility situations.
To forward the message, both RTS and CTS events need to be occurred successfully, and then the probability for the whole forwarding process, P_{FWD}, is computed as follows:

$$\begin{aligned} P_{FWD} &= Pr(CTS, RTS|\overline{Dest}, Rout) \\ &= Pr(CTS, RTS) \times Pr(RTS|\overline{Dest}, Rout) \\ &= (1 - P_{DROP}) \times [1 - (P_{COL} + P_{MOB})] \end{aligned} \tag{3.16}$$

Therefore, the probability of occurrence of packet dropping attack can be deduced using the following formula:

$$P_{DROP} = 1 - \frac{P_{FWD}}{[1 - (P_{COL} + P_{MOB})]} \tag{3.17}$$

Finally, the dropping probability is compared to a predefined detection threshold θ. If P_{DROP} is greater than this threshold, then the corresponding node is considered as malicious, and as legitimate otherwise.

3.2.4 Discussion

Packet dropping attacks were widely studied in many research works. In this section, many cryptographic techniques, trust management and classification frameworks were presented and discussed. We summarize these mechanisms in table 3.1. We first classify them in four categories: detection schemes, preventive schemes, security extensions and secure routing protocols. Secondly, we show the security enhancements provided by each mechanism in terms of availability, authentication, privacy and integrity. Finally, we evaluate their impact on network performance according to four parameters: end-to-end delay, overhead, packet delivery ratio and throughput.

Attributes			Security type				Security improvements				Performance drawbacks			
		Security mechanisms	Detective policy	Preventive policy	Security extension	Security protocol	Availability	Authentication	Privacy	Integrity	End-to-end delay	Packet delivery ratio	Overhead	Throughput
Cryptography solutions	Symmetric key cryptography	HSRBH [20]				✓		✓		✓	✓			
		TE [21]		✓					✓		✓	✓	✓	
		SEAODV [4]				✓		✓		✓			✓	
	Public key cryptography	ARAN [18]				✓		✓	✓	✓	✓		✓	
		AE [19]			✓				✓	✓	✓	✓		
		RRT & SRT [5]	✓						✓	✓	✓		✓	
		SAODV [17]				✓	✓	✓		✓	✓		✓	
Trust management	Reputation systems	Watchdog [22]		✓			✓	✓		✓			✓	
		CORE [23]				✓				✓	✓		✓	✓
		CONFIDANT [24]				✓	✓			✓			✓	✓
	Trust aware secure routing	TQR [7]			✓		✓						✓	
		TAODV [25]				✓	✓			✓			✓	
		FrAODV [26]			✓			✓		✓	✓	✓		
		FPNT-OLSR [52]			✓		✓				✓		✓	
Classification frameworks	FSM	LRI-based [27]		✓			✓			✓			✓	
	Dynamic learning	Anomaly detection [28]	✓				✓				✓			✓
	Trust prediction	TSR [29]				✓	✓			✓		✓		
	IDS	RTS&CTS [30]			✓		✓					✓		

Table 3.1: Classification of packet dropping security mechanisms.

As a global conclusion, we note that most of cryptography-based solutions [17] [4] [5], focus on the integrity of routing packets in order to prevent the malicious modification of the non-mutable fields of routing packets such as, SN and TTL or HC fields. The main drawback of these mechanisms is the significant amount of bandwidth they consume in terms of bandwidth and network overhead, which is

not suitable for resource-constrained networks (i.e., MANETs).

On the other hand, trust-based solutions [7] [25], introduced the notion of reputation in ad hoc routing; when the routes between communicating nodes are made up of trusted nodes. The trust value of a node is adjusted according to its behavior, namely, based on its cooperation in the packet forwarding process. Although these solutions force the malicious nodes to cooperate in routing, the trust value can be falsified if many colluding nodes give good recommendations about a malicious node; solutions that use indirect reputation like CONFIDANT [24] are susceptible to such attacks.

Finally, classification frameworks were also proposed to detect or prevent packet dropping attacks in MANETs. These solutions are usually based on a behavior monitoring process in order to detect protocol deviations. The performance of a these solutions depends on the selected attributes or attributes they use to make behavior classification. For instance, the solution proposed in [28] use RREQ and RREP packets as attributes, while that proposed in [30] analyzes the behavior of a network node based on RTS and CTS packets.

3.3 Mechanisms against flooding attacks

Flooding attacks occur usually when the underlying routing protocol uses a flooding-based method to disseminate routing information, such as AODV [16] and DSR [104]. In such routing protocols, a threshold value of RREQ generation rate is defined in order to prevent malicious flooding behaviors. Moreover, the maximum number of RREQ trials when a route request fails is also defined. However, these parameters still exploited by malicious entities, which can exhaust the nodes' resources by generating fake or malformed RREQ packets through the network. In addition, a malicious node can perform a data flooding after being included in paths established between communicating nodes. To protect MANETs against flooding attacks, many security mechanisms were proposed in the literature. In the following, we discuss some of research works that proposed solutions against this class of DoS attacks.

3.3.1 RREQ diffusion management

In a reactive protocol, network stills silent until a node has packets to send and there is no fresh route towards the intended destination. In this case, a flooding-based scheme is used to discover route by disseminating RREQ message among network nodes. Based on this characteristic, an authenticated attacker may exploit the RREQ flooding attack in the aim of exhausting network resources. In the following, we present some research works that were proposed to improve the RREQ dissemination scheme and alleviate the impact of flooding attacks.

Protocol and standard specifications

The existing countermeasures and prevention schemes in the literature focus mainly on RREQ flooding attacks such as, AODV RFC [16]. According to this standard, a node should not originate more than RREQ_RATELIMIT RREQ messages per second. In addition, after broadcasting a RREQ, a node waits for a RREP or other control message with current information regarding a route to the appropriate destination. If a route is not received within NET_TRAVERSAL_TIME milliseconds, the node may try again to discover a route by broadcasting another RREQ, up to a maximum of RREQ_RETRIES.
According to the DSR routing standard [104], an *exponential back-off* is used in order to limit the rate at which new route discoveries may be initiated by any node for the same destination. If a node attempts to send additional data packets to this same node more frequently than this limit, the subsequent packets should be buffered in the *Send Buffer* until a RREP is received, but it must not initiate a new route discovery until the minimum allowable interval between new route discoveries for this target is reached.
However, both specifications mentioned above are designed to manage the flooding based route discovery. Thus, from a security point of view, they cannot be adapted to prevent the occurring of flooding attacks. A *Distributed DoS* (DDOS) can be merely launched by some colluding nodes which respect the specified rate limit, will never be detected.

Rate Limitation

In [31], the authors proposed a technique to identify and isolate the malicious node that flood the network based on the *Anonymous Secure Routing* (ASR) protocol [107]. This protocol cannot differentiate the packets that are originating from a particular source node from the packets that are destined to a particular destination node. Thus, a node receiving a large number of packets from its previous hop node cannot determine whether it is being flooded by its previous hop node or by the nodes prior to its previous hop.
To deal with flooding attacks, the authors proposed the use of *rate limitation* component before being transmitted to the next hop neighbor. Every node monitors each requesting neighbor's channel usage at regular intervals. If the packets transmitted by the neighbor exceeds a certain *transmission threshold* within a given interval, then subsequent packets are dropped. If a neighbor generates requests with a rate exceeding the *transmission threshold* by some *blacklist threshold*, then it is believed to be responsible for flooding, and all the packets received from it are discarded in the future intervals.
If a blacklisted neighbor node exhibits a normal behavior for a *white-list* threshold, the monitoring node removes that neighbor from the blacklist and begins to forward packets for the neighbor. Using this method, every participating node is forced to share the transmission channel equally with the neighbors.

3.3.2 RREQ dropping

The purpose of this class of mechanisms is to identify the nodes generating an excessive amount of RREQ messages, and then forces them to fulfill the RREQ generation rate allowed by the protocol specification.

Address filtering

The aim of the filter proposed in this work [32] is to limit the rate of RREQ packets. Each node maintains two threshold values. The threshold values are the criterion for each node's decision of how to react to a RREQ message. A BLACKLIST_LIMIT parameter is introduced to specify a value that aids in determining whether a node is acting maliciously or not. If the RREQ message generation rate of a node exceeds this parameter value, a flooding attack is detected. Then, the blacklisted node is ignored for a time period given by BLACKLIST_TIMEOUT after which it is unblocked. By blacklisting a malicious node, a distributed prevention process is ensured by all neighbors of the malicious node by restricting the RREQ flooding.
On the other hand, if the generation rate of RREQ messages is between the RREQ_RATELIMIT and BLACKLIST_LIMIT values, the RREQ packet is added to a *delay queue* waiting to be processed. Every DELAY_TIMEOUT, the first packet in the queue is removed to be processed. Therefore, a malicious node having a high RREQ rate is severely delayed.

Random Assessment Delay (RAD)

The authors of [33] presented a security analysis of RREQ flooding attacks and their impact on network performance, especially the network throughput. They proposed this mechanism to keep track of redundant RREQ packets received over a defined period. Each node monitors and counts the RREQ messages it receives from each RREQ sender during a defined time period.
At the end of the time period, the node computes the rate at which it has received the RREQ packets from each sender. To distinguish between malicious RREQ floods and those generated by normal nodes, they calculate a cut-off rate. The RREQ messages from a sender having a smoothed average rate which is greater than the cut-off rate will be dropped without forwarding.
The authors argue that this technique has no adverse impact in the absence of malicious control packet floods, but stops any harmful effects of frequent control packet floods without the need to identify the malicious nodes.

Flooding Attack Prevention (FAP)

FAP is an adaptive statistical packet dropping mechanism proposed in [34] to defend against RREQ flooding and data flooding attacks using two methods: *neighbor suppression* and *path cut-off*.
In this mechanism, the authors assume that the isolation of a flooder node can be

ensured by its neighbors by rejecting the packets it sends. In addition, they proposed the use of a *priority-based* packet processing method instead of the standard method used in AODV, which is *First In First Out* (FIFO). The priority value of a node is assigned by its neighbors; an inversely proportional relation is defined between the RREQ generation rate and the priority value. A node having a priority value less than a certain threshold is considered as attacker, and then isolated by its neighbors.
The path cut-off method was proposed to delete an established path between an attacker and a victim node. The authors proposed simply to send a RERR message in order to break the links containing the attacker. However, they have not presented how an attacker can be detected, they only proposed a method to stop the flooding attack. Moreover, sending a RERR may break legitimate links, and then downgrade the packet delivery ratio.

3.3.3 Trust-based flooding detection

The core idea of these approaches is to evaluate the trust of a node and then assign a reputation value based on its behavior. To detect flooding attacks, the trust computation takes into account the rate of packets generated by network nodes. In the following, we present two security mechanisms that were used trust and reputation systems to detect flooding attacks.

Dual Defensive Wall System (DDWS)

Jiang *et al.* [35] proposed this system to reduce the disruption caused by flooding attacks and mitigate their negative impact on network resources and the route discovery process. Their approach aims to detect two aspects of RREQ flooding attacks: the massive RREQ dissemination and the bogus route discovery.
The first line of defense is provided by neighbors of a RREQ generator to verify if the destination IP address belongs to the network. The second line of defense is ensured by the destination node which detects the large number of RREQ packets. Using two threshold values of RREQ generation $Min\_Threshold$ and $Max\_Threshold$, nodes can be classified according to three priority levels: *normal*, *gray* and *black*. A node with priority 1 has a frequency less than $Min\_Threshold$, and with priority 3 has a frequency more than $Max\_Threshold$. The list of suspect nodes having priority equal to 2 are those having a frequency between $Min\_Threshold$ and $Max\_Threshold$ values. The downgrade and upgrade policies between the priority levels are defined according to the changes of the RREQ generation frequency.

Friendship-based flooding detection

This approach was proposed in [36] to detect flooding attacks based on the extent of friendship among network nodes. Nodes are categorized as *friends*, *acquaintances* or *strangers* based on their relationships with their neighboring nodes. Any new node entering the network is considered as stranger to all its neighbors.

The mutual trust level of two communicating nodes can be upgraded to acquaintance and later to friend if the number of successful packet transmissions becomes higher than defined thresholds.
For each level of trust a maximum number of RREQ and data packets is also defined which can be used to detect nodes trying to generate malicious requests and/or send useless data traffic to the network.
If the specified threshold level is reached, further RREQ or data packets received from the initiating node are ignored and dropped. This approach uses the trust value as a parameter in order to prevent flooding attacks. However, there is no clear description of the method used to "compute" the trust values.

3.3.4 Anomaly prevention systems

These prevention systems were proposed to monitor the activity of a network node, and then detect any type of misuse that falls out of normal protocol specifications.

Anti-flooding AODV (AF-AODV)

Similarly, a monitoring of the RREQ transmission rate is proposed in [37] to detect RREQ flooding attacks using a mechanism called *AF-AODV*. During the monitoring period *TrafficTime* if the rate exceeds a threshold then the neighbor is added to a black-list, and the RREQ packets are not forwarded but they are still recorded.
A black-listed node is suspended if the rate continues to be high, and can be removed if the rate drops below the threshold. If a node finds that many neighbors exceed this threshold, then the *Black_list* value associated to the nodes generating the greatest number of RREQ is set to 1 and the other neighbors are suspended.
If the value of *Black_list* is greater than or equal to 1, then the node tests the authenticity of the neighbor by replying with a fake RREP packet.
If the neighbor is malicious, this will not result in any data flowing and its *Black_list* is incremented. Otherwise (i.e., if it is legitimate) data will flow to the fake RREP packet originator which can respond with a RERR packet so that a new route can be found.
This process is executed each time a RREQ packet is received from a black-listed node until a value equal to 5, beyond which that node is considered as malicious and is excluded from the neighbor list.

Specification-based Intrusion Detection (SIDE)

Panos *et al.* [38] proposed this detection engine to safeguard the operation of the AODV routing protocol. The authors proved that the anomaly-based approaches are not suitable for dynamic environments such as MANETs. They proposed a system that ensures a real-time monitoring of local information in order to detect malicious activities that try to violate the legitimate functionality of AODV. An FSM model is defined for each message processing operation in AODV in order to prevent deviations from the standard functionality.

To detect RREQ flooding attack, SIDE monitors the generation of RREQ messages at each network node and detect whether a host node attempts to perform RREQ flooding attack. This is achieved by validating the originator IP address encapsulated in the generated RREQ and by monitoring if the rate of RREQs per second exceeds the RREQ rate limit.
Besides the SIDE engine, a remote attestation procedure is used to enable a node to verify if a particular neighboring node operates an untampered version of SIDE. This can ensure the authenticity of each node running a SIDE instance and guarantee its integrity.

3.3.5 Discussion

Despite the severity of flooding attacks, there are only few research works studying their impact or proposing appropriate security solutions to prevent them. In figure 3.1, we summarize the categories of security solutions against these attacks that we discussed in this section.

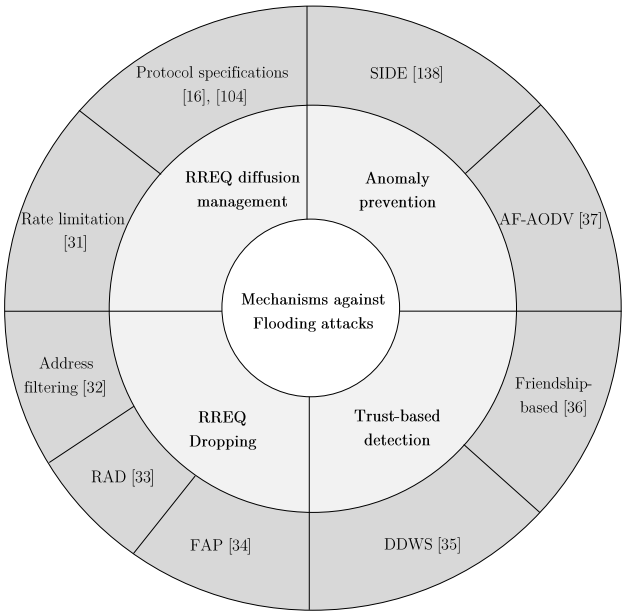


Figure 3.1: Security mechanisms against flooding attacks in MANETs.

Most of the existing proposed solutions handle the problem of RREQ flooding, which can be occurred in flooding-based routing protocols such as, AODV and DSR. The prevention of such attacks is challenging, especially because they can be launched by external malicious nodes. In other words, an attacker can send an excessive number of RREQ to a network node without being in the network.
Since the protocol improvements are not sufficient to prevent these attacks, many research works were investigated to mitigate the impact of RREQ flooding attacks,

and detect the nodes performing such misbehavior.
The prevention approaches proposed in [31] and [33] use similar methods to detect and exclude nodes performing a RREQ flooding attack. They rely on a monitoring process of the rate of RREQ packets generated by network nodes to detect the occurrence of an abnormal traffic of requests, and then the detected nodes are temporally suspended or definitively excluded from the network.
Other works proposed trust-based approaches to adjust the permitted generation rate of RREQ packets for each node according to its trust level such as, [36] and [34]. However, these approaches cannot prevent an attacker from falsifying the rate of generated RREQ packets and then disrupt the trust assessment process.
Finally, to ensure an efficient prevention of these attacks, it is important to improve the reliability and security of flooding-based routing algorithms based on specification similar to the approach proposed in [38]. These improvements should protect MANETs against resource exhausting attacks, which helps to maintain the availability of network services.

3.4 Mechanisms against routing disruption attacks

Protecting MANETs against routing disruption attacks and ensuring secure neighborhood creation is an extremely important issue. In this section, we review some research works that handle the problem of malicious traffic attraction and tunneling attacks, which are usually launched before a modification or a packet dropping attack. Broadly, the different detection mechanisms may fall into four categories: *time-based*, *hardware-based*, *connectivity-based* and *statistical-based* mechanisms.

3.4.1 Time-based mechanisms

In this class of security mechanisms, the detection of Wormhole nodes is based on traversal time and hop count analysis. The aim of these mechanisms is to ensure that transmitting nodes lie within the local neighborhood according to the routing protocol specifications.

Temporal leashes

This work was proposed by Hu *et al.* in [39] to enable a receiver of a packet to determine if a packet has traveled further than the *leash* allows. To use temporal leashes, the sending node includes in the packet the time at which it sent the packets. When receiving a packet, the receiving node compares this value to the time at which it received the packet, and then it is able to detect if the packet traveled too far based on the claimed transmission time. To implement this mechanism, the authors use the *Time Efficient Stream Loss-tolerant Authentication* (TESLA) with *Instant Key* expiration (TIK).
The intuition behind TIK is that the packet transmission time can be significantly longer than the time synchronization error. The authors argue that the TIK protocol

provides protection against the Wormhole attack, since an attacker that retransmits the packet will most likely delay it long enough that the receiver will reject the packet, because the corresponding key has already expired and the sender may have disclosed it.
However, knowledge of the positions of all nodes may be a prerequisite for correctly estimating transmission times. Moreover, the authentication data needed to protect packet leashes requires additional processing and communication overhead.

TrueLink

Eriksson *et al.* [40] proposed this mechanism as an extension to the IEEE 802.11 MAC layer in order to prevent the creating of Wormhole tunnels in MANETs. Using TrueLink, a node i can verify the adjacency of an apparent neighbor j, using a combination of timing and authentication. First, the nodes exchange nonces α_j and β_i. This exchange proves the adjacency of the responding node through the use of strict timing constraints; only a direct neighbor is able to respond in time.
Second, i and j transmit a signed message $(\alpha_j\ ,\beta_i)$, mutually authenticating themselves as the originator of their respective nonce. To implement Truelink, the authors showed how a nonce can be included in each CTS frame without changing the frame format using the IEEE 802.11 standard.

Delay Per Hop Indication (DelPHI)

This mechanism was proposed in [41]; it relies on collecting information about HC and delay of disjoint paths between a source and a destination in order to verify whether a certain path is subjected to Wormhole attacks.
In this approach, the authors consider that the delay a packet experiences in propagating one hop under normal situation should be similar along each hop along the path. However, under a Wormhole attack, the delay for propagating across false neighbors should be high since there are in fact many hops between them.
Based on this attack signature, the delay per hop of a legitimate path is compared with the delay per hop of a path that is under Wormhole attack. Therefore, if a path has a ratio delay per hop exceeding some threshold value, it is likely to be subjected to a Wormhole attack.

Transmission time analysis

In [42], a Wormhole detection mechanism based on the *Round-Trip-Time* (RTT) was proposed for the *Ad hoc On-demand Distance Vector* (AOMDV) routing protocol. In this mechanism, the *transmission time* is evaluated between every two successive nodes in an established path between a source S and a destination D. After obtaining the RTT values, S calculates the *total RTT* for established routes. The detection of Wormhole has some originality since the authors propose a solution adapted to AOMDV routing protocol, where S estimates a threshold value t_{max} of RTT by calculating the maximum time taken for a packet to traverse 1-hop. t_{max} is

estimated by averaging the time elapsed to receive an acknowledgement after sending a Hello message for all neighbors of S.
When S has many routes towards a destination D, it can check for an established route r if it contains Wormhole nodes by comparing its total RTT with an estimated RTT denoted as t_e. This latter is calculated as follows: $t_e = t_{max} \times h$, where t_{max} is the estimated time to traverse r, and h is the hop count of r. A Wormhole node is detected in r if its total RTT is greater than the estimated value t_e.

3.4.2 Localization-based mechanisms

This class of mechanisms use location information in conjunction with temporal information in the aim of detecting shortcuts caused by Wormhole nodes. In the following, we discuss some of these solutions, and we describe for each solution how the spatio-temporal information is exploited to detect Wormhole attacks in MANETs.

Geographical leashes

Geographical leashes were introduced in the same work that proposed the use of temporal leashes to detect Wormhole attacks [39]. In this approach, each node must know its own location, and all nodes must have tightly synchronized clocks. The core idea consists of appending the location information of the sending nodes to each packet, and verify whether the hop-by-hop transmission is physically possible. When a node sends a packet, it includes in the packet its own location, and the time at which it sent the packet. A node receiving this packet compares these values to its own location, and the time of receiving of packet. Then the receiver computes an upper bound on the distance between the sender and itself. According to the obtained distance, the receiver is able to detect if the packet has traveled further than the leash allows, and then detect the tunnels created by Wormhole nodes.

Secure Tracking Of node encounters (SECTOR)

The SECTOR mechanism was proposed in [43] to prevent Wormhole attacks without requiring any clock synchronization. The proposed mechanisms in this work aim to ensure a *secure verification* of the time encounters between nodes in the aim of preventing Wormhole attacks in MANETs.
A *Mutual Authentication with Distance* bounding (MAD) protocol is used to apply the same principle as packet leashes, with the difference that each node can perform distance bounding without having to trust on the other party.
The technique used by MAD consists of a series of rapid bit exchanges between nodes. Each bit sent by the first node is considered to be a *challenge* for which the other node is required to send a one bit response immediately. By measuring the time between sending out the challenges and receiving the responses, the first node can compute an upper-bound on the distance to the other node.
Despite the fact that this mechanism does not need a clock synchronization, it

requires a special hardware to measure the local timing with nanosecond precision, which is impractical for MANETs.

Directional antennas

The use of this technique was introduced in [44] as an attempt to mitigate Wormhole attacks by equipping network nodes with directional antennas. This solution is based on the assumption that a Wormhole attack cannot be performed when the neighbor sets are maintained correctly. Their approach is based on three protocols: *directional*, *verified* and *strict* neighbor discovery.
The directional neighbor discovery protocol forms the basis of two other protocols; it ensures the authentication of each node in its neighborhood. Then, the messages received from nodes that are not members of its neighbor set are ignored. The verified neighbor discovery protocol defines the steps required to authenticate the nodes and their apparent relative positions by using specific nodes called *verifiers*. This protocol ensures that the directions towards two communicating nodes are consistent. The strict neighbor discovery protocol is similar to the verified discovery protocol, however, it adds some conditions on the possible location of the verifier nodes between neighbor nodes. This restriction can prevent a smart adversary trying to convince two slightly out of radio range nodes that they are neighbors.

3.4.3 Connectivity-based mechanisms

Since Wormhole attacks attempt to create malicious shortcuts between colluding nodes, they have a noticeable effect on the network from a geometry point of view. Another signature of this attack, is the fact that the Wormhole node attracts neighbors by advertising itself as having a fresher route to an intended destination. In the following, we present mechanisms that rely on network structure analysis to detect Wormhole attacks.

Geometric Random Graphs

Lazos *et al.* [45] presented the necessary and sufficient conditions for detecting and defending against Wormhole attacks, using geometric random graphs induced by the communication range constraint of nodes. In this solution, it is assumed that the existence of Wormhole links violates the geometric graph model, by allowing links longer than the normal transmission range, thus transforming the initial geometric graph into a logical connectivity graph, where arbitrary connections can be established. In other words, a link is considered as a Wormhole if the distance between its two endpoints exceeds the regular communication range. Besides this mathematical modeling, a set of trusted specialized guards are also employed to prevent local nodes from launching a Wormhole attack using a global pre-loaded key in the nodes.

LiteWorp

A lightweight countermeasure against Wormhole attack called LiteWorp was proposed in [46]. The detection process is composed of three phases: *building neighbor lists*, *local monitoring* and *isolation algorithm*.
LiteWorp starts with building a data structure of the first and second hop neighbors. First hop neighbor information is used by a node to reject (respectively do not forward) packets that are received from a node (respectively to a node) that is not a neighbor.
Also, the second hop neighbor information is used to verify if a forwarded packet comes from a neighbor of the forwarder. Once the neighbor list building is completed, a node starts a local monitoring to detect Wormhole attack and diagnoses the malicious nodes involved in launching it. The monitoring of malicious activities is ensured by a set of *guard* nodes based on a *watch buffer*; the outgoing and incoming links of each node are monitored by these guards, which are in fact its 1-hop neighbors.
Finally, a *malicious counter* is maintained at each guard node and incremented each time a malicious activity is detected. If the counter of a node X exceeds some threshold, the corresponding guard node sends an alert message indicating the detection of a malicious node. Then, each node receiving the alert message removes X from its neighbor list.
Among the limitations that can be identified in this work is its inability to determine neighbors at arbitrary points in the lifetime of the network. Another challenge arises from the possibility for a mobile attacker to perform malicious actions at one location and moves.

Control Traffic Tunneling Attack Countermeasure (CTAC)

This mechanism was proposed in [47] to prevent malicious nodes claiming to exist in more than one location in the network. CTAC relies on a set of trusted nodes which called *Cluster Head* (CH) for position tracking of the mobile nodes and bookkeeping of adversarial behavior of a mobile node. Depending on the number of available CH nodes, CTAC divides the network into geographical units. Each geographical unit is assigned to a CH and contains all the network nodes that lie within that unit. Two types of detection were defined in CTAC:

- **Local detection:** the malicious node is detected by the nodes in its current neighborhood in a distributed fashion.
- **Global detection:** the malicious node is detected on a global network scale by the CH nodes in two phases.
    1. *Reports aggregation* within the geographical unit of CH at multiple locations.
    2. *Reports exchange* between CHs about the nodes that move from one geographical unit to another.

In CTAC, the mobile node can send and receive its own traffic but cannot forward any traffic. This design arises from the insight that a node can only launch a Wormhole attack if it is allowed to relay packets.
Using this approach, the isolation of malicious nodes is achieved in two phases; locally, whereby the malicious node is removed from the current neighborhood, and globally using global information at the CH nodes, and then a peripatetic mobile node cannot cause unbounded damage in the network.

3.4.4 Anomaly detection techniques

Many research works were proposed to detect the anomaly caused by Wormhole attacks in MANETs. Most of them rely on statistical methods to evaluate the deviation from a normal model by investigating certain classification attributes. In the following, we present three of research works that used statistical analysis methods to detect the Wormhole attacks in MANETs.

Neighbor distribution

Buttyan *et al.* [108] proposed two mechanisms for Wormhole detection in WSNs which only require a list of neighbors for each node.
The first mechanism is based on the fact that the adversary increases the number of neighbors of nodes within their transmission range by introducing new links into the network graph. Then, the hypothetical and real distribution of the number of neighbors are compared using a *Neighbor Number Test* (NNT) to detect the existence of a Wormhole.
The second detection mechanism is based on the fact that Wormhole attack distorts the distribution of the length of the shortest paths among all pairs of nodes. This mechanism consists also of a comparison between a hypothetical and the real distribution of the length of shortest paths among all pairs of nodes.
The main drawback of this mechanism is that it only focuses on detecting the presence of Wormhole nodes, but they do not pinpoint their locations.

Statistical Analysis of Multi-path routing (SAM)

This mechanism was proposed in [109]; it is based on the observation that certain statistics of the discovered routes by routing protocols will change under Wormhole attacks. The authors assume that the tunneled link established between the colluding nodes is extremely attractive to routing requests, and then it is expected that the majority of the obtained routes will contain that link.
To detect Wormhole attacks in a set of obtained routes R, a requesting node computes the relative frequency of each link that appears in R, and deduces the maximum relative frequency p^{max}. On the other hand, it computes the difference ϕ between the p^{max} link and the second most frequently appeared link in R from the same route discovery.
Finally, the authors deduces that both statistics p^{max} and ϕ are much higher in the

presence of malicious tunnels than that in normal system. Therefore, these statistics are used in conjunction to determine whether the routing protocol is under Wormhole attack.

Statistical Wormhole Apprehension using Neighbors (SWAN)

This scheme was proposed in [110] which uses a localized statistical approach by taking advantage of the nodes' mobility. The authors assume that the *neighborhood count* is expected to increase beyond a range of statistical fluctuation when a node encounters with a Wormhole attacker. This fluctuation can be quantified by computing the difference between the history of neighborhood counts S_{train}, and the recent sample set S_{test}.
The SWAN maintains the most recent events of the number of neighbors using *kernel sliding windows* for regular training W_{reg} and test W_{test}. When a node is not in the Wormhole state, it checks if the normalized distance between S_{train} and S_{test} distributions is greater than a threshold T_{sh}, then it concludes that it entered into a Wormhole, and the new W_{test} value is added to a Wormhole state sliding window W_{wh}. Otherwise, it updates the regular sliding window W_{reg} entries by removing the oldest elements and adding new entries with W_{test}.
When a node is already in a Wormhole state, and the distance between the distributions of S_{train} and S_{test} becomes less than the threshold T_{sh}, the node concludes that is not in a Wormhole state any more. In this case, the W_{wh} entries are cleaned up, and the W_{test} value is used to update the sliding window class W_{reg}. Otherwise, it updates the W_{wh} entries by removing the oldest elements and adding new entries with W_{test}.

3.4.5 Discussion

In this section, we have described some security solutions that were proposed to mitigate, prevent or detect Wormhole attacks in MANETs, which are summarized and classified in figure 3.2.

Among the discussed security solutions, time-based mechanisms can be divided into two classes according to the strategies they use: clock synchronization and RTT. The tightly synchronized clocks needed in the case of packet leashes [39] are not adapted to MANETs due to the packet delays that may vary caused by the unpredictable network mobility. On the other hand, the detection mechanisms based on RTT [42] eliminate the need for tight clock synchronization required in temporal leashes.

Localization-solutions use geographical location information to detect shortcuts caused by Wormhole attacks. The assumption of these solutions such as geographical leashes [39], is that each network node knows its exact location,

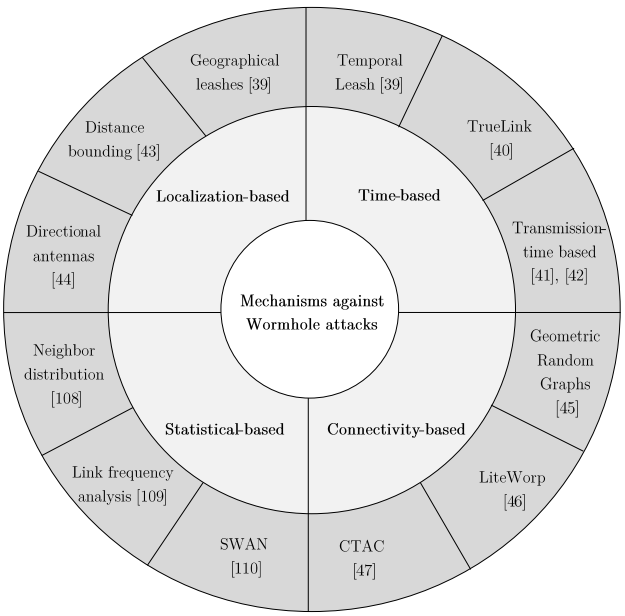


Figure 3.2: Security mechanisms against Wormhole attacks.

and embeds the location and a time-stamp in each packet it sends. However, such mechanisms cannot be adapted in MANETs, because they rely on location information, which is dynamically changing.

Connectivity-based solutions are generally based on an analysis of topological structure to detect anomalies caused by Wormhole attacks in the network. To detect malicious shortcuts launched by Wormhole nodes, these solutions take into consideration geometric attributes such as, established links between nodes, logical graph models and neighbor nodes distribution. However, the consistence of attributes is always related to the network mobility, and then the use of such solutions is challenging in MANETs.

Finally, the statistical based solutions are easy to integrate with intrusion detection systems to secure routing, especially in the case of on-demand protocols. However, the main drawback of these solutions is that the routing anomaly can be detected as long as sufficient information of routes is available.

3.5 Conclusion

Throughout this chapter, we have reviewed some security mechanisms against DoS attacks in MANETs. We presented them within three axis according to the attack class they handle: packet dropping, flooding attacks and routing disruption.
More focus was noticed on defense mechanisms against packet dropping attacks in MANETs. We have discussed three classes of research works that handle this

problem: cryptography-based, trust-based and classification-based solutions.
In the following, we clarify the research considerations that will be further developed in the rest of this thesis. First, we intend to address the problem of DoS caused by packet dropping attacks, by taking into consideration the efficient techniques used to launch such attacks. Many existing research works handling this problem have not presented an in-depth modeling of this class of attacks, and then the detection accuracy still an intrinsic challenge. In other words, an inappropriate modeling of dropping attack may produce a confusion with the legitimate packet dropping caused by packet collusion or link congestion. To address this problem, we propose in the next chapter a fully decentralized framework to detect packet dropping attacks in MANETs based on a probabilistic classification of nodes' behaviors.

Chapter 4

Behavior-based detection against ad hoc routing misbehaviors

"It isn't that they can't see the solution. It is that they can't see the problem."

– *Gilbert K. Chesterton*

Contents

4.1 Introduction

In the previous chapter, we have presented many research works that were proposed to protect MANETs against routing attacks. The availability of routing services is an important security requirement, which cannot ensured using cryptographic tools alone. In fact, the use of a monitoring mechanism such as, Watchdog [22] enables a node to collect information about its neighbors' behavior using a network configuration called promiscuous mode. The collected information is supplied to a misbehavior detection process which ensures the availability of routing services by excluding misbehaving nodes from the network.
The Watchdog mechanism has the advantage of having neither additional traffic nor significant computational overhead. However, when using Watchdog, nodes need to store many information to perform misbehavior detection. This latter should be lightweight in order to optimize the exchanged data traffic especially when the network is dense. In addition, the selection of attributes required to recognize the nature of a node's behavior affects the accuracy of the detection process. Thus, normal and attack models must be well defined in order to make the rate of false alerts as low as possible.
Finally, the definitive isolation of malicious nodes is not recommended for self-organized networks like MANETs, especially if a legitimate node is declared as malicious due to a faulty detection. Excluding a node from the network means that the network looses a routing entity, and then the routing performance is degraded. Thus, it is useful to give a chance to an excluded node to participate later in routing if it behaves legitimately; this ensures a trade-off between security and QoS.

In this chapter, we present an efficient detection scheme against packet dropping attacks that threat the routing layer in MANETs. The detection is performed in a hop-by-hop fashion, in order to allow a node to select the next hop node providing a secure route, not only a shorter or faster route when forwarding packets towards an intended destination.
Taking into account the aforementioned security requirements, the detection phases of our proposed framework can summarized as follows:

- First of all, we propose the use of a monitoring process at each network node in order to collect information about packets exchanged in its transmission range. We are interested by three packet types to model the behavior of a monitored node: Data, RREQ and RREP. The collected information about exchanged packets is modeled as a vector composed of three elements representing the forwarding rates of each packet type.

- After modeling the behavior of a neighbor, a classification process is performed using Bayes theorem, in order to evaluate the probability of maliciousness of a node based on its behavior vector. To detect packet dropping attacks, we use the Bernoulli and Multinomial Bayesian models. In Bernoulli model, the vector elements are booleans representing the occurrence or absence of packet

dropping. However, in the case of Multinomial model, the vector attributes consider the number of packets transmitted by a node during the monitoring period.

- Based on the probability of maliciousness of a behavior vector, a node is detected as malicious if its probability value exceeds the classifier threshold, and classified as legitimate otherwise. This process can be used to ensure reliable and secure routing decisions in MANETs. Thus, when a node has packets to forward, it should avoid neighbors detected as malicious, and use those nodes satisfying security requirements.

In terms of implementation, the proposed detection framework is not designed to be used as a security extension for an existing routing protocol. However, it can be used as a host-based detection system at each network node in order to avoid malicious nodes when forwarding packets. The simulation results show a good performance of our proposed framework, thanks to the accurate misbehavior detection provided by the Bayesian classification models.
The chapter is organized as follows. We introduce in section 4.2 the notations used in the chapter, and then discuss the related work in section 4.3. In section 4.4, we introduce the attack model handled in this chapter, and the design of the behavior-based probabilistic approach that we propose to detect misbehaving nodes in MANETs. Section 4.5 is dedicated for an extensive description of the two Bayesian models that we use for node classification: Bernoulli and Multinomial models. In section 4.6, we validate the performance of these models and make a comparative study of their efficiency through simulations. Finally, we conclude the chapter in section 4.7.

4.2 Notations

In the following, we introduce the notations that are used in this chapter in their appearing order.

Notation	**Meaning**
S	a source node
D	a destination node
X	a network node
t	a timestamp
τ	the duration of a training period
t_0	the beginning time of a bootstrap phase
Δt	the duration of a time interval
ER	an expected packet forwarding ratio
AR	an acquired packet forwarding ratio
pt	a packet type

$\overrightarrow{x}$	a binomial behavior vector
X_i	a binomial random variable associated to a packet type pt_i
ϑ	the space of possible behaviors x
p	a probability value
c	a class of behavior
l	a legitimate node
m	a malicious node
T_{ref}	the reference table associating x and p values
T_N	a table storing p values of neighbor nodes
N_i	a Multinomial random variable associated to a packet type pt_i
M_{ref}	the reference matrix composed of Multinomial probabilities $p(X_i/c)$
$\overrightarrow{\eta}$	a Multinomial behavior vector
W^b_{err}	the referential weighted error of a filter
W_{err}	the weighted error of a filter
TP	the rate of true positive
FP	the rate of false positive
TN	the rate of true negative
FN	the rate of false negative
TCR	the total cost ratio of a filter
n_0, $n_0^{'}$	rate of malicious nodes in the network

4.3 Related work

In this section, we discuss the existing research works that were proposed to detect malicious behaviors in MANETs, and we show how we exploit the approaches proposed in these works to design the phases of our detection framework.

The first interesting research work is that proposed in [111]. It consists of a semi-decentralized *Intrusion Detection System* (IDS) to detect misbehaving nodes in MANETs. The core idea of this approach is to deploy several IDS nodes in the network in order to detect and prevent packet dropping attacks. IDS nodes are set in promiscuous mode in order to sniff all routing packets within their transmission range and estimate a suspicious value of other nodes according to the amount of abnormal difference between RREQ and RREP messages they exchange. When the suspicious value exceeds a predefined threshold, an attack is detected and a block message is broadcast to all nodes to cooperatively isolate the malicious node.

The second work that we consider is proposed in [112]. This work is not dedicated for MANETs, however, it uses a Bayesian classification scheme that we find interesting to design our detection framework. The contribution of this work is a performance evaluation of three classification models in the context of Spam detection: Naive, Multinomial and Bernoulli Bayesian models. The core idea is to calculate the probability that an email is spam given that the email contains

a defined word. According to Naive Bayes, a spam detection is ensured based on the probability that a document is spam given that a set of words occur in the document. Multinomial Bayes is an optimization of Naive Bayes that is used to make the filter more accurate by keeping track of the number of occurrences of each word, namely, the number of times that a word takes place in a multiset of words. Multivariate Naive or Bernoulli Bayes classifies an email based on boolean attributes representing the presence or the absence of a word instead of its occurrence.
Based on research works that we described above, we exploit the attribute selection of two types of *control packets*, RREQ and RREP, as proposed in [111], and we consider *data packets* as an additional attribute to model the behavior of a node.
On the other hand, we adapt the Bernoulli and Multinomial Bayesian filters presented in [112] to classify the behavior of a network node. We evaluate the behavior of a node based on the obtained probability of maliciousness, in order to verify if it performs a packet dropping attack.

4.4 System design

In this section, we present the general design of the mechanism that we propose to detect packet dropping attacks in MANETs. Basically, the detection phases of the proposed mechanism are fully decentralized, where each network node runs an instance of the detection algorithm in order to avoid misbehaving nodes when it has packets to forward. In the following, we present the network and system assumptions, the characteristics of dropping attack which is handled in the proposed solution and the different modules composing our detection mechanism.

4.4.1 Assumptions

In this thesis we consider the case of homogeneous networks, where all nodes are identical in terms of battery energy and hardware complexity. We assume that all nodes have the same transmission range and use an omni-directional antenna. In addition, we suppose that the promiscuous mode is available to all network nodes; a forwarding node will be heard by other nodes in its transmission range.
Regarding the storage, we assume that each node is able to store information about messages exchanged by its neighbors. The proposed scheme is non-cryptographic; neither additional module nor a-priori key distribution is required. In terms of processing, we consider that the stored information can be analyzed at each node using an algorithm that we present throughout this chapter.
We use AODV as the base routing protocol to apply our detection mechanism. Being a source routing protocol, where each node can easily recognize the entire packets passing by its neighbors, which is appropriate to ensure the behavior monitoring process.
Finally, we assume the existence of *reference tables* that we obtain during a training

phase and maintained by each network node during a bootstrap phase starting at $t = t_0$. These tables are used in the classification models to calculate the probability of maliciousness and detect misbehaving nodes. The use of these tables will be detailed throughout this chapter.

4.4.2 Attack model

We already showed in 2.3.3 the data packet loss caused by packet dropping attacks in MANETs. In this section, we aim to refine the description of these attacks by presenting their techniques and objectives.
To represent the different dropping attack types handled in our study, we define three elementary dropping events associated to three packet types:

- **Data packet dropping**: when an attacker is included in an established route and drops a part or whole data packets it receives instead of forwarding them to the appropriate node.
- **RREQ packet dropping**: by dropping this type of packets, the attacker can disrupt the normal discovery process launched by a requesting node to find out a route to a requested node.
- **RREP packet dropping**: an attacker may maliciously prevent the establishment of a route by not forwarding the reply message from a requested node towards a requesting node.

Based on these elementary events, an attacker may perform a dropping of one or more packet types according to its malicious goals. For instance, if the attacker is already included in an established route, it can drop data packets in order to disrupt the normal packet delivery between two communicating nodes.

4.4.3 Detection specification

As we already mentioned, the proposed detection mechanism is fully decentralized, which is adapted to infrastructure-less nature of MANETs. Each node should maintain an instance of this mechanism in order to detect and avoid misbehaving neighbors. The proposed detection process evaluates the probability of maliciousness of 1-hop neighbor nodes based on their behaviors, and then avoid those performing packet dropping attacks.
The detection process is composed of four phases as illustrated in figure 4.1. This process is performed periodically, and the phases depend on each other as follows:

1. **Periodic monitoring**: each node is set in a promiscuous mode in order to listen to the exchanged packets within its transmission range during a period Δt. The outcome of this monitoring is an information about the number of packets exchanged by each neighbor node.

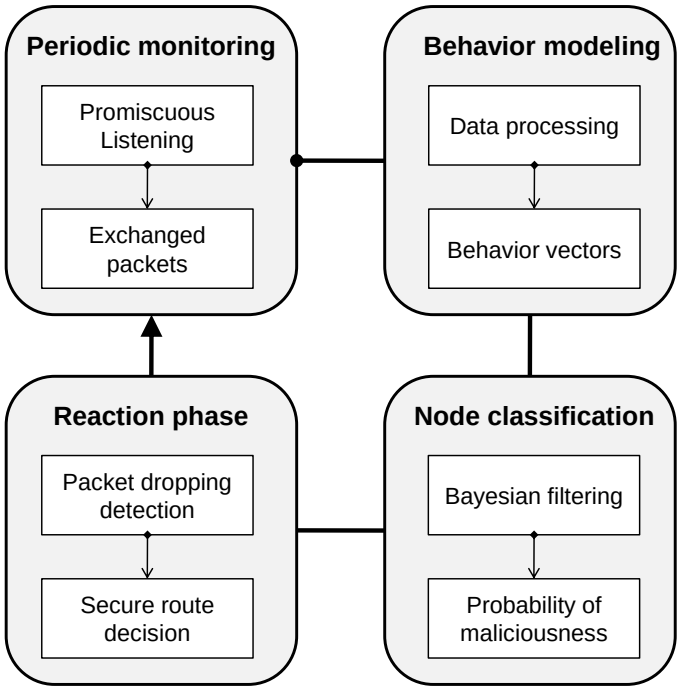


Figure 4.1: Modules of the proposed detection framework.

2. **Behavior modeling**: based on the collected information during the monitoring phase, the behavior of each neighbor is modeled as a vector composed of three attributes. Each attribute corresponds to a packet type and represents the rate of packets that are successfully forwarded by the neighbor node. We use binomial and multi-class vector attributes, which are adapted to the classification models that we use in our detection mechanism.

3. **Node classification**: a node can calculate the probability of maliciousness of a given vector using Bayesian classification. By comparing the obtained probability value with the classifier threshold, the behavior can be classified as legitimate or malicious.

4. **Reaction phase**: according to the class of a neighbor's behavior, a node may perform two possible reaction events. If the behavior vector is classified as legitimate, the node can use that neighbor as a next hop for its future packet forwarding. Otherwise, the neighbor node should be avoided.

This probabilistic approach is not implemented neither as a real detection system, nor a secure routing protocol, it is validated through network simulations. It is noticed that the eventual purpose of this detection process is to allow a node to recognize the behavior of its neighbors, and then make secure routing decisions.

4.5 Attacks classification mechanism

As we already mentioned, the behavior of a node is modeled according to three monitored routing events: data forwarding, route discovery and route establishment, which are respectively associated to Data, RREQ and RREP packet types. The behavior vector is classified according to its probability of maliciousness, which is calculated according to Bayesian theory.
The Bayesian probability theory can be defined as a transformation from the *prior*, $P(H)$, to the *posterior*, $P(H|D)$, formally reflects what has been learned about the validity of the hypothesis from consideration of the data according to the following relation [113]:

$$P(H|D) = \frac{P(D|H) \times P(H)}{P(D)} = \frac{Likelihood \times Prior}{Evidence} \tag{4.1}$$

Where $P(D|H)$ is the *likelihood* function assessing a probability of an observed data D arising from a hypothesis H, and $P(D)$ is the *evidence* obtained by integrating $P(D|H) \times P(H)$ over all H.
This relation defines the Naive Bayes rule, which really involves nothing more than the manipulation of conditional probabilities. Among the optimization of Naive Bayes, we are interested by the Bernoulli and Multinomial models, that we adapt to handle the problematic of packet dropping attacks in MANETs.
In this section, we explain the mathematical definition of these models, and we refine their application in our detection mechanism.

4.5.1 Bernoulli classification model

Bernoulli models constitute a class of exact Bayesian filters for non-linear/non-Gaussian recursive estimation of dynamic systems, recently emerged from the random set theoretical framework. The Bernoulli filter is the optimal Bayes filter for a single dynamic system which can randomly switch *on* or *off* [114]. We present in the following the mathematical definition of the Bernoulli model, the structure of behavior vectors used as classification input and the probabilistic scheme that we exploit to detect misbehaving nodes in MANETs.

4.5.1.1 Notations and preliminaries

Bernoulli distribution is a discrete probability distribution of a random variable X which is given as follows:

$$p\,(X = x) = \begin{cases} p & if \;\; x = 1, \\ 1 - p & if \;\; x = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Meaning that X takes the value 1 with probability p and 0 with probability $q = 1-p$, which is equivalent to $p(X = x) = p^x(1-p)^{1-x}1_{\{0,1\}}(x)$. The average and variance

are given by the formula: $E(X) = p, Var(X) = p(1-p)$.
According to this definition, the behavior of a node can be characterized by a vector $\overrightarrow{x}$ defined by $\overrightarrow{x} = (x_1, \ldots, x_m)$. The vector elements $x_1, \ldots, x_m$ are the values taken by the random variables $X_1, \ldots, X_m$ that are assumed conditionally independent given category c, which can be legitimate or malicious. Each variable gives information about forwarding of a specific packet type.
The random variables are binary, where each X_i is set to 1 if packets having the type i denoted by pt_i are correctly forwarded, and 0 otherwise. Consequently, each random variable X_i follows a Bernoulli distribution with a parameter $p_i = p(pt_i)$.

4.5.1.2 Binomial behavior modeling

We define a normal behavior as a set of events performed by a network entity to ensure the forwarding of packets according to the routing protocol specifications. This can be written as following:

$$AR_{pt} \approx ER_{pt}\,, \quad \forall\ pt \in S,\ where\ S = \{Data,\ RREQ,\ RREP\} \tag{4.2}$$

where pt is a packet type, AR is the *Acquired Ratio* of forwarded packets by a node and ER is the *Expected Ratio* of packets that must be forwarded. For unicast packets, namely, Data and RREP packets, the forwarding ratio is computed using the following formula:

$$Forwarding\ Ratio = \frac{\#\ of\ forwarded\ packets}{\#\ of\ packets\ excpected\ to\ be\ forwarded} \tag{4.3}$$

In the case of RREQ packets, which are disseminated in broadcast mode, AR and ER metrics are calculated similarly using the following formula:

$$Disseminating\ Ratio = \frac{\#\ of\ disseminated\ packets}{\#\ of\ packets\ excpected\ to\ be\ disseminated} \tag{4.4}$$

ER metric is estimated for each packet type during a training phase by taking into account the network setup and other parameters such as, link congestion and packet collusion. The AR metric is calculated during a test phase, and compared with the ER value in order to model the node's behavior and verify if there is a deviation from the normal behavior model.
Based on ER and AR values, we model the behavior of a node as a vector which is composed of three boolean attributes $\vec{x} = (x_{Data}, x_{RREQ}, x_{RREP})$ and can be obtained as follows:

$$x_p = \begin{cases} 1 & if \quad AR_{pt} < ER_{pt} \\ 0 & if \quad AR_{pt} \geq ER_{pt} \end{cases} \tag{4.5}$$

Consequently, for each packet type, there are two possible events: 1 if the packet is properly forwarded, and 0 otherwise. Hence, the finite space of possible behaviors ϑ has $2^3 = 8$ vectors which are:

$$\vartheta = \{(0,0,0); (0,0,1); (0,1,0); (0,1,1); (1,0,0); (1,0,1); (1,1,0); (1,1,1)\} \tag{4.6}$$

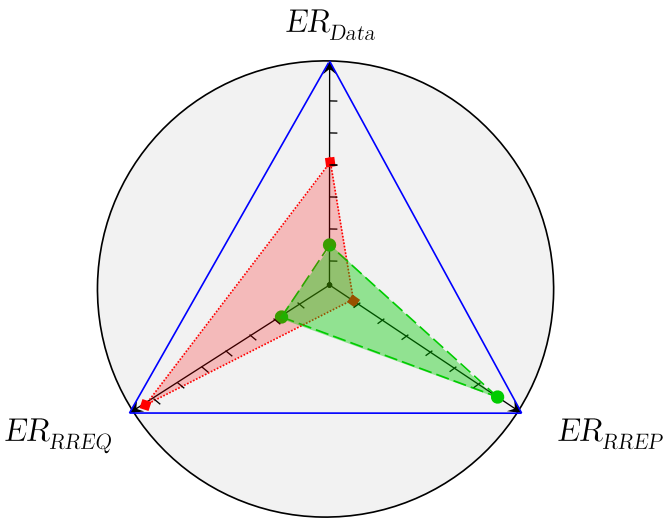


Figure 4.2: Packet dropping attack patterns.

To refine this scheme, we illustrate in figure 4.2 a graphical representation of the possible node's behavior models. The blue triangle represents a triplet of ER values, and the red and the green ones represent triplets of AR values for two dropping attack models. For instance, AR_{Data} and AR_{RREP} are respectively less than ER_{Data} and ER_{RREP}, which can be represented by the behavior vector $\vec{x_1} = (0, 1, 0)$. The green triangle is another attack pattern with AR_{Data} and AR_{RREQ} having values less than ER_{Data} and ER_{RREQ} respectively, which can be represented by the vector $\vec{x_2} = (0, 0, 1)$.

4.5.1.3 Node classification

According to the Bayes theorem [115] and the total probabilities theorem, for a node $(x_{Data}, x_{RREQ}, x_{RREP})$, the probability to belong to class c is defined by:

$$p\left(C = c/\overrightarrow{X} = \overrightarrow{x}\right) = \frac{p(C = c) \times p(\overrightarrow{X} = \overrightarrow{x}/C = c)}{p(\overrightarrow{X} = \overrightarrow{x})} \tag{4.7}$$

In our case, we define two possible classes of node's behavior, l and m, which stand respectively to *legitimate* and *malicious*. Based on this definition and using the theorem of the total probability [116], we deduce:

$$p\left(C = c/\overrightarrow{X} = \overrightarrow{x}\right) = \frac{p(C = c) \times p(\overrightarrow{X} = \overrightarrow{x}/C = c)}{\sum_{c \in \{l,m\}} p(C = c) \times p(\overrightarrow{X} = \overrightarrow{x}/C = c)} \tag{4.8}$$

We assume that X_1, X_2, and X_3 random variables are conditionally independent given class c. As we already mentioned, $S = \{pt_1, pt_2, pt_3\}$ is the set of packet types constituting a behavior vector. Using the set of packet types, the probability that

a behavior $\vec{x}$ belongs to class c expressed in 4.8 becomes:

$$p\left(C = m/\overrightarrow{X} = \overrightarrow{x}\right) = \frac{\prod_{i=1}^{3} p(pt_i/m)^{x_i} \times (1 - p(pt_i/m))^{(1-x_i)} \times p(m)}{\sum_{c \in \{l,m\}} \prod_{i=1}^{3} p(pt_i/c)^{x_i} \times (1 - p(pt_i/c))^{(1-x_i)} \times p(C = c)} \quad (4.9)$$

When classifying a behavior using this probabilistic scheme, there are two types of errors that may be occurred. The first possible error is classifying a malicious node as legitimate, and the second one is classifying a legitimate node as malicious, which are denoted respectively by $m \to l$ and $l \to m$. In our case, we consider that the second error is more costly than the first one, in order to minimize the rate of false alerts that may be obtained when detecting malicious nodes. To represent the rate between these two errors, we introduce a cost parameter called λ. For instance, using $\lambda = 5$, the $l \to m$ error is five times more costly than the $m \to l$ one. Based on this assumption, a node having a behavior vector $\vec{x}$ is classified as malicious if:

$$p\left(C = m/\vec{X} = \vec{x}\right) > \lambda \times p\left(C = l/\vec{X} = \vec{x}\right) \quad (4.10)$$

Since $p(C = m/\vec{X} = \vec{x}) + p(C = l/\vec{X} = \vec{x}) = 1$, we can deduce that the node $\vec{x}$ is classified as malicious if and only if:

$$p\left(C = m/\vec{X} = \vec{x}\right) > \alpha, \ where\ \alpha = \frac{\lambda}{1+\lambda}\ and\ \lambda = \frac{\alpha}{1-\alpha} \quad (4.11)$$

According to this classification criteria, if the probability of maliciousness denoted by $p(C = m/\vec{X} = \vec{x})$ exceeds α, then the node having a behavior $\vec{x}$ is classified as malicious. If the two mentioned errors have the same cost, namely, $\lambda = 1$, then the threshold of classifier denoted by α is set to 0.5. In our case, we use many values of λ associated to classification threshold α that we defined in 4.11, in order to evaluate the performance of the used classification models (see section 4.6).
We present in the following the detection process implementing the Bernoulli classification, in order to detect misbehaving neighbor nodes when forwarding packets.

4.5.1.4 Application of Bernoulli filter

The detection becomes effective after the running of a bootstrap phase, in which each node keeps a reference table denoted by T_{ref}. This latter is obtained during a training phase, and associates for each vector $\vec{x}$ in the behavior space ϑ, a probability of maliciousness value $p(\vec{x})$ calculated according to equation 4.11.
Table 4.1 shows an association between behavior vectors and probability values which are computed by averaging many network simulation results that we extracted from NS2 trace files. This table allows a node to evaluate the maliciousness of a given behavior by looking up its associated probability value. In addition, at each node, a neighbor table T_N is used to maintain, for each neighbor node, an entry

Table 4.1: Reference table for misbehavior detection.

Behavior model $\vec{x}$	Probability $p(\vec{x})$
000	0.971
001	0.687
010	0.507
011	0.622
100	0.375
101	0.242
110	0.130
111	0.099

storing its ID, the probability value p and a timestamp t.

Further, the information stored in T_N is used to verify the reliability of a node before using it to forward packets. The entry of a neighbor node is updated based on periodic calculation of probability values. If a neighbor node goes out the transmission range of monitoring node, its corresponding entry is removed from T_N.

4.5.2 Multinomial classification model

In the case of Binomial distribution, we can imagine that $B(n, p)$ is obtained by considering a sequence of n independent draws in the game of heads or tails, and p as the probability of drawing head. However, in the case of Multinomial distribution, we generalize the binomial distribution, where each "run" can produce k different results, not only two. For example, we can imagine a sequence of n tosses of a "dice" with k sides, side number i with probability p_i out. In the following, we present the mathematical definition of the Multinomial Bayesian model, and we explain how it can be adapted as a behavior-based attacks detection mechanism in MANETs.

4.5.2.1 Multinomial distribution

A Multinomial distribution is characterized by n (e.g., the number of shots) and the sequence $(p_1, p_2, \ldots, p_m)$ with $p_1 + p_2 + \cdots + p_m = 1$. Following n shots, we denote n_i the number of shots producing the result number i, $n_1 + n_2 + \cdots + n_m = n$. The shots are random, and the n_i are the achievements of m random variables that we denote by N_i, where $i = 1, 2, \ldots, m$. These variables are not independent since they are related by $\sum_i N_i = n$.

We call Multinomial distribution $Mult(n, p_1, p_2, \ldots, p_m)$ the joint distribution of the k random variables N_i. It is therefore a multivariate discrete distribution. Its support is all k-uples of positive integers or zero $(n_1, n_2, \ldots, n_m)$ such that,

$n_1 + n_2 + \cdots + n_m = n$.
The distribution $Mult(n, p_1, p_2, \ldots, p_m)$ is entirely determined by the values of probabilities of each possible m-uples. These probabilities are denoted by $P(N_1 = n_1, \ldots, N_m = n_m)$ and are given as follows:

$$p\left(N_1 = n_1, \ldots, N_m = n_m\right) = n! \times \prod_{i=1}^{n} \frac{p_i^{n_i}}{n_i!} \tag{4.12}$$

For all m-uples belonging to the support of the distribution, and 0 otherwise. The average, variance and covariances are given by the formula:

$$E(N_i) = np_i, Var(N_i) = np_i(1 - p_i), Cov(N_i, N_j) = np_ip_j \tag{4.13}$$

4.5.2.2 Multi-class behavior modeling

In this statistical model, each node is characterized by a vector $\overrightarrow{\eta}$ defined by $\overrightarrow{\eta} = (n_1, \ldots, n_m)$, where $n_1, \ldots, n_m$ are the values taken by the random variables $N_1, \ldots, N_m$ that are assumed conditionally independent given class c. In our case, there are three random variables N_1, N_2 and N_3, which are associated to forwarding rates of three packet types: Data, RREQ and RREP.
Similarly to the modeling scheme that we used in the case of Bernoulli distribution in 4.5.1.2, each vector attribute is associated to a packet type. However, in the case of Multinomial, each attribute provides an information about "*how much packets are forwarded*" instead of the "*packet is forwarded or not*" information. Hence, an attribute n_i represents the percentage of packets pt_i that are forwarded. For a packet type pt_i, the associated attribute n_i can be obtained using the following formula:

$$\%\ of\ forwarded\ packets = \frac{\#\ of\ forwarded\ packets}{\#\ of\ packets\ excpected\ to\ be\ forwarded} \times 100 \tag{4.14}$$

In the case of RREQ packets, the value of n_i is associated to the percentage of disseminated packets, since they are generated using a broadcast mode. Hence, n_i is similarly obtained as follows:

$$\%\ of\ disseminated\ packets = \frac{\#\ of\ disseminated\ packets}{\#\ of\ packets\ excpected\ to\ be\ disseminated} \times 100 \tag{4.15}$$

Since the Multinomial Bayesian theory relies on the attribute occurrence, we assume that the percentage is truncated as integer value. For instance, a vector having the following form: $\overrightarrow{\eta} = (97, 30, 10)$, means that 97% of Data packets, 30% of RREQ packets, and 10% of RREP a packets are forwarded.

4.5.2.3 Node classification

With a Multinomial event model, vector attributes represent the frequencies with which certain events were generated by a Multinomial $(p_1, \ldots, p_n)$. Hence, for a

node $(n_{Data}, n_{RREQ}, n_{RREP})$, the probability to belong to class c can be defined as a generalization of the binomial law defined in formulas 4.7 and 4.8, and obtained as follows:

$$p\left(C{=}c/\overrightarrow{N}{=}\overrightarrow{\eta}\right){=}\frac{p(\overrightarrow{N}=\overrightarrow{\eta}/C{=}c)\times p(C{=}c)}{p(\overrightarrow{N}{=}\overrightarrow{\eta}/C{=}c)\times p(C{=}c){+}p(\overrightarrow{N}=\overrightarrow{\eta}/C{=}c)\times p(C{=}\bar{c})} \quad (4.16)$$

As we already mentioned, the input vectors of the Multinomial model are the occurrences of packet transmission having types pt_i. Taking into account the types and percentages of forwarded packets by a node $\overrightarrow{\eta}$, and based on the formula 4.12 we obtain the following formula:

$$\begin{aligned} p\left(\overrightarrow{N}{=}\overrightarrow{\eta}/C{=}c\right) &= p((N_1,\ldots,N_m){=}(n_1,\ldots,n_m)/C{=}c) \\ &= p((N_1{=}n_1,...,N_m{=}n_m)/C{=}c) \\ &= n!\times\prod_{i=1}^{n}\frac{p(pt_i/C{=}c)^{n_i}}{n_i!} \end{aligned} \quad (4.17)$$

By combining the formulas 4.16 and 4.17, we obtain the following formula to calculate the probability that a node $\overrightarrow{\eta}$ belongs to class c:

$$p\left(C{=}c/\overrightarrow{N}{=}\overrightarrow{\eta}\right){=}\frac{\prod_{i=1}^{n}p(pt_i/C{=}c)^{n_i}\times p(C{=}c)}{\prod_{i=1}^{n}p(pt_i/C{=}c)^{n_i}\times p(C{=}c)+\prod_{i=1}^{n}p(pt_i/C{=}\bar{c})^{n_i}\times p(C{=}\bar{c})} \quad (4.18)$$

In our case, c and $\bar{c}$ stand respectively to malicious and legitimate behavior classes. By applying the formula 4.18, the probability of maliciousness of a node $\overrightarrow{\eta}$ can be obtained as follows:

$$p\left(C=m/\overrightarrow{N}=\overrightarrow{\eta}\right)=\frac{\prod_{i=1}^{3}p(pt_i/m)^{n_i}\times p(m)}{\prod_{i=1}^{3}p(pt_i/m)^{n_i}\times p(m)+\prod_{i=1}^{3}p(pt_i/l)^{n_i}\times p(l)} \quad (4.19)$$

By simplifying the equation 4.19, we obtain the following formula:

$$p\left(C=m/\overrightarrow{N}=\overrightarrow{\eta}\right)=\frac{1}{1+\frac{\prod_{i=1}^{3}p(pt_i/l)^{n_i}\times p(l)}{\prod_{i=1}^{3}p(pt_i/m)^{n_i}\times p(m)}} \quad (4.20)$$

where $p(pt_i/m)$, $p(m)$ and $p(l)$ are the estimated frequencies calculated on a learning corpus. Consequently, the selection criteria is similar to that used in Bernoulli model but applied conditionally to the event $(\overrightarrow{N}=\overrightarrow{\eta})$. Therefore, a node $\overrightarrow{N}$ is classified as malicious if $p(C=M/\overrightarrow{N}=\overrightarrow{\eta})>\alpha$, where the threshold value α is associated to the cost value λ defined in 4.11.

4.5.2.4 Application of Multinomial

In the case of Bernoulli classification, we have modeled the behavior of a node as a vector of booleans, where each vector attribute provides an information about the occurrence or the absence of a dropping of each packet type pt_i. However, in the case of Multinomial model, the vector attribute is multi-class, it represents the percentage of forwarded packets for a packet type pt_i.
To recognize the legitimacy or the maliciousness of a multi-class behavior vector, we introduce a 2×3 reference matrix M_{ref}. Each element of M_{ref} is associated to a Multinomial probability $p(X_i/c)$, where X_i is a random variable corresponding to packet type $pt_i \in \{Data, RREQ, RREP\}$, and $c \in \{malicious, legitimate\}$. The elements of M_{ref} are calculated using the following formula:

$$p\left(X_i/c\right) = \frac{\#\ of\ packets\ X_i\ transmitted\ in\ c}{\#\ of\ transmitted\ packets\ in\ c} \tag{4.21}$$

The elements of this matrix represent the Multinomial probability distribution for each packet type during a training phase. In the following, we show an example of a reference matrix M_{ref} that we can obtain through simulations:

$$M = \begin{pmatrix} 0.531 & 0.401 & 0.068 \\ 0.230 & 0.375 & 0.395 \end{pmatrix}$$

The first line of M represents the probability distribution of packets transmitted by legitimate nodes, while the second line represents the probability distribution of packets transmitted by malicious nodes. Based on elements of matrix M, we can calculate for a given behavior vector the probability of maliciousness according to formula 4.20.

4.6 Performance evaluation

In this section, we prove through simulation the ability of the Bayesian models proposed in this chapter to detect malicious nodes in MANETs. First, we describe the experimental setup of simulation scenarios that we have performed. Then, we present the algorithms we use to implement dropping attacks, analyze trace files and model the node behaviors. Finally, we present the application of the used Bayesian filters, and we evaluate and compare their performance using different filtering threshold configurations.

4.6.1 Simulation environment

The performance of Bernoulli and Multinomial models regarding packet dropping attacks detection in MANETs is evaluated through simulation. We use a well-known discrete event simulator Network Simulator 2 (NS2) to perform simulations of a mobile ad hoc network in the presence of a changing percentage of malicious nodes.

The attack models defined in 4.4.2 are implemented by modifying the original C^{++} source files of AODV routing protocol. Then, a genuine version of AODV is executed by legitimate nodes, while a malicious one is executed by the attackers. The experimental parameters of the simulated MANET are shown in table 4.2.

Table 4.2: Network simulation parameters.

Parameter	**Value**
Coverage area	800 m × 800 m
Number of nodes	$15 - 50$ nodes
Transmission range	250 m
Simulation time	$5 - 15$ minutes
Mobility model	Random Waypoint
Antenna	OmniAntenna
Maximum speed	15 m/s
Routing protocol	AODV
Traffic type	UDP/ CBR
% of attackers	$5, 10, \ldots, 40\%$
MAC layer type	IEEE 802.11p

The attackers perform packet dropping according to the function that we created in the malicious version of AODV. Many scenarios were performed using different network sizes ranging from 15 to 50 nodes. The percentage of attackers in the network is incremented by 5% from 5% to 40%. At the end of simulation scenarios, the trace files are maintained to be filtered and analyzed using data processing algorithms, which are presented in the following.

4.6.2 Trace files processing

The output of simulations that we performed using NS2 is a trace file describing the events performed by each node in the network. These files have usually a size between 200 MB and 350 MB when the simulation time is about 20 minutes. To extract an interesting information from such large files, we have developed data processing algorithms using GAWK language.
The obtained information consists of statistics on exchanged packets by each network node during the simulation. As we already mentioned throughout this chapter, the packet types that we handled in our study are: Data, RREQ and RREP packets.
Based on these statistics, we affect for each node a vector modeling that represents its behaviors, use those vectors as inputs for probability calculation phase and then classify them according to the obtained probability value.

4.6.3 Node classification process

The data set of nodes that we obtained after modeling the behavior vectors is exported as an input file of the algorithm that we implemented using MATLAB. This algorithm allows a node to calculate the probability of maliciousness of a given vector, and then gives the appropriate class according to the filter threshold α.
For each Bayesian filter, we developed a MATLAB function, which takes a behavior vector as input and returns a probability of maliciousness value. In the case of Bernoulli filter, we use the reference table T_{ref} that we obtained after a training phase. This table associates for each behavior vector in the finite space ϑ a probability value as we already explained in section 4.5.1.4. However, in the case of Multinomial classification, we use the reference vector V_{ref} introduced in section 4.5.2.4.
The obtained results associate for each network node a probability of maliciousness value, which is calculated using the Bernoulli and Multinomial Bayesian filters. Therefore, each node in the dataset is labeled as malicious or legitimate according to the probability value compared to the used classifier threshold α. In the following, we present the parameters that we use to evaluate the performance of the filters, and then we make a comparative analysis of their efficiency using different configurations.

4.6.4 Cost sensitive evaluation measures

After labeling the nodes of the dataset using Bayesian filters, we evaluate the performance of each filter by comparing their error rate with a "baseline". This latter defines the case where no filter is used: legitimate nodes are (correctly) never detected, and malicious nodes (mistakenly) always pass the filter.

We define the *weighted error* parameter W_{err} to evaluate the error when using a filter, and a *referential weighted error* W^b_{err} to evaluate the error without using the filter [117]. These parameters are defined as follows:

$$W_{err} = \frac{\lambda FP + FN}{\lambda N_l + N_m}, \quad W^b_{err} = \frac{N_m}{\lambda N_l + N_m} \tag{4.22}$$

where:

- TP: *True Positive*, denotes the number of legitimate nodes classified correctly.
- FP: *False Positive*, denotes the number of legitimate nodes classified mistakenly.
- TN: *True Negative*, denotes the number of malicious nodes classified correctly.
- FN: *False Negative*, denotes the number of malicious nodes classified mistakenly.
- $N_m = FN + TN, \quad N_l = TP + FP$

To compare the performance of a classification model with that of the baseline, we introduce a new parameter called *Total Cost Ratio* (TCR), which allows us to calculate the ratio between their error rates W_{err} and W^b_{err} as follows:

$$TCR > 1 \Leftrightarrow W^b_{err} > W_{err}$$

When selecting the detection threshold of the filter, we assume that $l \rightarrow m$ is λ times more costly than $m \rightarrow l$. Based on this assumption and using the formula 4.22, we obtain the following formula to calculate the TCR value:

$$TCR = \frac{W^b_{err}}{W_{err}} = \frac{N_m}{\lambda FP + FN} \tag{4.23}$$

Greater TCR values indicate better performance. When the value of TCR is greater than 1, the error rate of the baseline is greater than the error rate when using the filter. In this case, the classification is considered as interesting since it minimizes the error rate. Otherwise, when the TCR is less than 1, the error rate when using the filter is greater than the error rate of the baseline. In this case, the baseline is better. Based on this measure, we evaluate the performance of Bayesian filters, and then we discuss their ability of detecting packet dropping attacks in MANETs.

4.6.5 Simulation results

As we already mentioned in 4.6.4, the default classification threshold α is equal to 0.5, when the cost of false positive ($l \rightarrow m$) is equivalent ($\lambda = 1$) to that of false negative ($m \rightarrow l$).

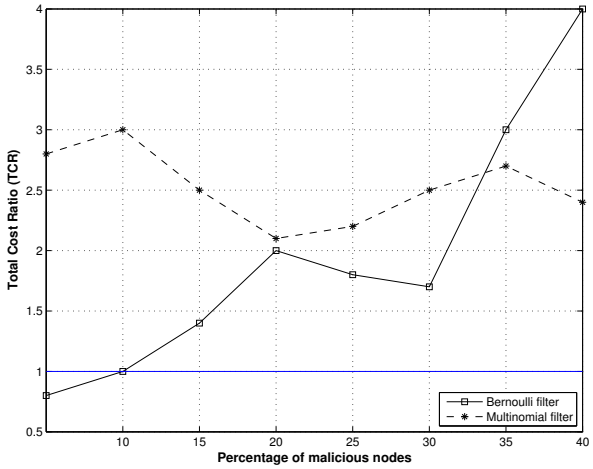


Figure 4.3: $\lambda = 1$ and $\alpha = 0.5$

Using this configuration, figure 4.3 shows that Multinomial model has a better performance than Bernoulli model when the proportion of malicious nodes in the network is less than 33%. At this proportion, the Bernoulli model has a better performance than Multinomial filter. However, when the proportion of misbehaving

nodes in the network is less than 10%, the TCR value of Bernoulli model becomes less than 1, and then the baseline is better.

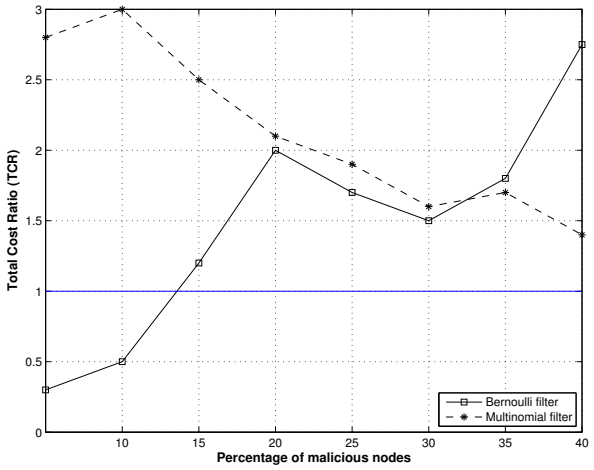


Figure 4.4: $\lambda = 3$ and $\alpha = 0.75$

In a second configuration (figure 4.4), the cost between the two errors λ is set to 3, and then the classifier threshold is equal to 0.75. In this case, the TCR curve in the case of Multinomial filter has a decreasing shape when the proportion of malicious nodes increases. However, it stills always better than Bernoulli filter, since it never has a $TCR < 1$. Similarly to the first configuration of λ, Bernoulli filter shows a better performance when the proportion of malicious nodes exceeds 33%.

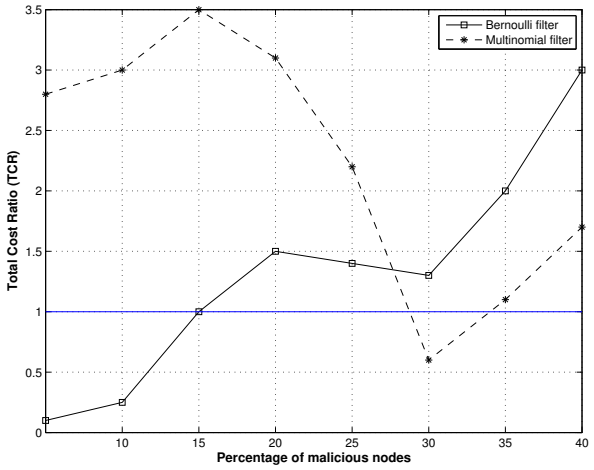


Figure 4.5: $\lambda = 9$ and $\alpha = 0.9$

When the false positive error is 9 times more costly than the false negative error ($\lambda = 9$), the classifier threshold is equal to 0.9. In this case, Bernoulli filter has no interesting classification when the percentage of malicious nodes is less than 15%.

Beyond this percentage, the TCR curve of Bernoulli filter has an increasing shape, on the contrary of Multinomial one, which has a decreasing shape and becomes without any interesting filtering when the percentage of malicious node is between 27% and 34%.

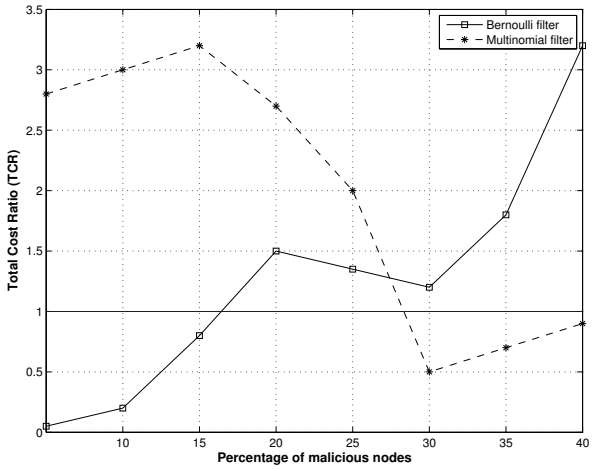


Figure 4.6: $\lambda = 19$ and $\alpha = 0.95$

Finally, when we set the cost between the two errors to $\lambda = 19$, the classifier threshold becomes $\alpha = 0.95$. Similarly to the previous case, with a low percentage of malicious nodes (15%), the Bernoulli filter is not interesting, otherwise, it has a $TCR > 1$ whatever the percentage of malicious nodes in the network. On the other hand, the Multinomial filter has a bad performance when the percentage of malicious nodes exceeds 27%.

4.6.6 Discussion

Based on the performance evaluation presented in the previous section, we deduce that the TCR curve in the Bernoulli case is above the linear equation $TCR = 1$ (i.e., the baseline) when the number of malicious nodes is greater than certain percentage n_0, and that n_0 increases when λ increases. Therefore, more the parameter λ increases, the size of the network to which the Bernoulli model is not worth increases. On the other hand, in the Multinomial case, we note that the TCR curve is above the baseline except when λ is set to 9 or 19 where the curve is below the baseline with a certain percentage of malicious nodes denoted by $n_0^{'}$. The Bernoulli model performs well when λ is set to 1 or 3 regardless the percentage of malicious nodes, and when λ is set to 9 or 19 while the percentage of malicious nodes is less than $n_0^{'}$ which is between 25% and 30%. As a comparison between the two models, the TCR curve of the Multinomial model is above that of the Bernoulli model until a percentage $n_0^{'}$ beyond which the positioning of the curves is inverted.
We already showed that the Bernoulli filter uses vectors providing information about "forwarding" or "not forwarding", while Multinomial model uses the forwarding rate

itself which provides more information on how much packets are forwarded. This latter is a supplementary information that makes the Multinomial classification model able to recognize the low proportions of malicious nodes, and further it can ensure a full detection when λ is set to 0.5. On the other hand, in the case of Bernoulli classification, the boolean attributes providing information about the occurrence or the absence of dropping attack is sufficient when the proportion of malicious nodes in significant, and ensures a better performance when the classification threshold is set to 0.5.
Based on this interpretation, we deduce that the choice of the appropriate filter is adjusted according to the proportion of malicious nodes in the network. We also conclude that a combination of the two models is interesting to ensure a full detection of dropping attacks regardless their percentage in the network.

4.7 Conclusion

In this chapter, we proposed a behavior-based probabilistic approach to detect packet dropping attacks in MANETs. The main goal of this contribution is to allow a node to avoid routes including misbehaving nodes by verifying the legitimacy of its 1-hop neighbors.
The rationale behind the proposed detection framework is to recognize the behavior of a node by evaluating the probability of being malicious based on its behavior. To calculate the probability value, we used a Bayesian classification which has been widely employed for spam detection. We selected two Bayesian filters, Bernoulli and Multinomial. In Bernoulli model, the probability is calculated based on the presence or absence of dropping of a packet type. However, in the case of the Multinomial model, the probability is calculated based on the rate of packets transmitted successfully.
Simulation results showed an efficient detection of packet dropping attacks using the Bayesian classification. A full detection of these attacks can be guaranteed by combining the advantages of the two models.
In the next chapter, we handle the problem of a sophisticated version of packet dropping attacks, in which a malicious node alternates its behavior by dropping packets sometimes and behaving normally other times. This kind of attacks is known as periodic dropping, which is difficult to detect due its similarity to normal packet discarding caused by certain network situations. To handle this problem, we propose to extend our detection mechanism by tracking the evolution of a monitored node over a long time period in order to predict their future state.

Chapter 5

Nodes' behaviors prediction through a stochastic analysis

"The only difference between a problem and a solution is that people understand the solution."

– *Charles Kettering*

Contents

5.1 Introduction

MANETs are dynamic networks composed of mobile entities which are free to move, join or leave the network without any constraint. In such decentralized environment, routing function is productive unless all network nodes in packet forwarding and other services.
As we already showed in chapter 3, most of existing routing protocols are not designed with security in mind and often are vulnerable to many attacks. An authenticated node can easily intercept an established communication session between legitimate nodes, and then drop the totality or a part of packets that are supposed to be forwarded.
In this chapter, we treat the problem of periodic packet dropping attacks, in which a malicious node randomly discards some packets during a time period. Obviously, the detection of such attacks is challenging, since it may be confused with legitimate dropping events which are happening usually due to packet collision or link congestion [118]. To address the aforementioned problem, we exploit two related research works to build the core idea of our proposed solution.
The first work is proposed in [30], in which the authors taken into account the legitimate packet discards to ensure an accurate identification of packet dropping attacks. They assumed that the legitimate packet dropping may be caused exclusively by *packet collision* and *link break*. Then, the malicious nodes can be detected by calculating the *probability of occurring a packet dropping* based on five attributes: number of sent and received data packets, number of sent RTS packets, number of received CTS packets and a verification of diffused RREQ.
In the second research work [119], the authors presented a decentralized trust inference model to protect MANETs against packet dropping attacks. Basing on the interest of historical behaviors of an entity, multi-dimensional trust attributes are incorporated to reflect trust relationship's complexity in various angles. The trust computation of a monitored node is based on direct experience of the monitoring node and a second hand information obtained using the recommendations of other nodes. By making use of the obtained *historical trust data sequence*, the authors proposed a Markov chain prediction model in order to provide a relative identification between normal and malicious behaviors and predict the future behavior of a node.
We believe that the approaches proposed in both research works are vulnerable to periodic packet dropping attacks. In the first work, the authors did not consider the RREP packets when calculating the probability of occurring a packet dropping. Hence, the malicious nodes discarding RREP packets cannot be detected. On the other hand, being a trust-based solution, the approach proposed in the second work is vulnerable to bad mouthing attacks [48]. For instance, if a malicious node provides a bad recommendation about a monitored node, the second hand information used in trust aggregation becomes incorrect. Therefore, the identification of behaviors of a node which is based on its trust value may be inconsistent.

To ensure an accurate detection of periodic dropping attacks in MANETs, we propose a novel prediction framework combining the advantages of the probabilistic behavior classification and the prediction model already presented. First, we perform a behavior analysis to calculate periodically the *probability of maliciousness* of a node by taking into account information about sent and received data, RREQ and RREP packets.
On the other hand, we perform a tracking process of the evolution of node's behaviors in the aim of addressing the problem of periodic dropping attacks. First, we associate a *maliciousness level* defining the behavior state of a node during a time period based on its probability value. Then, the *sequence of transitions* between these states over certain number of time periods are modeled using a Markov chain model. Finally, the stationary state of network nodes can be predicted, and then nodes performing a periodic dropping attack can be accurately detected.
The chapter is organized as follows. We introduce in section 5.2 the notations used in the chapter. Section 5.3 presents the problem of periodic dropping attacks in MANETs. In section 5.4, we describe the general design of solution components and we detail each of which in section 5.5. Simulation results and performance evaluation are presented in section 5.6. Finally, Section 5.8 concludes this chapter.

5.2 Notations

In the following, we introduce the notations that are used in this chapter in their appearing order.

Notation	**Meaning**
N_n	the number of network nodes
T	a time period
τ	a time slot
A	a monitoring node
B	a monitored neighbor node
$\overrightarrow{X}_\tau$	a behavior vector of a node during a time slot τ
pt_i	a packet type
x_{pt}	a boolean variable indicating whether packet pt is forwarded or not
C_τ	the class of a node's behavior during τ
l	a legitimate node
s	a suspicious node
m	a malicious node
p	a probability value
α_{min}	the left bound of suspiciousness interval
α_{max}	the right bound of suspiciousness interval
I_s	the suspiciousness interval
pr	a fuzzy proposition
R	a fuzzy rule

E	the space of node behavior states
k	a network node
t	a time moment
X_t	the state of the network at time t
X_t^k	the state of node k at time t
i, j	states of a node
P^k	the transition matrix of node k during T
n	the power of a transition matrix
$P^{k,n}$	the n^{th} power of transition matrix P^k
$p_{i,j}^k$	the probability that node k moves from state i to state j
$Tr_{(i,j)}$	the number of times the node k moved from state i to state j
C_i	the number of times the node k has visited the state i
$\nu_i^k(t)$	the probability that node k being in state i at time t
$\nu^k(t)$	the stochastic vector of node k
$\lim_{n\to\infty} \nu^k(n)$	the limit probability distribution of node k at the n^{th} power
hc	a hop count

5.3 Periodic dropping attacks in MANETs

Packet dropping attack is considered as one of the most severe DoS attacks threatening the ad hoc routing services. In this chapter, we treat the problem of *periodic dropping attacks*, in which a malicious node randomly discards some packets over a time period in order to appear as legitimate and deceive the underlying detection systems. The main challenge of these attacks is the fact that are closely similar to legitimate packet discards caused by packet collision or link congestion. Therefore, a malicious node performing such an attack may still a long period in the network and drop packets silently without being detected.

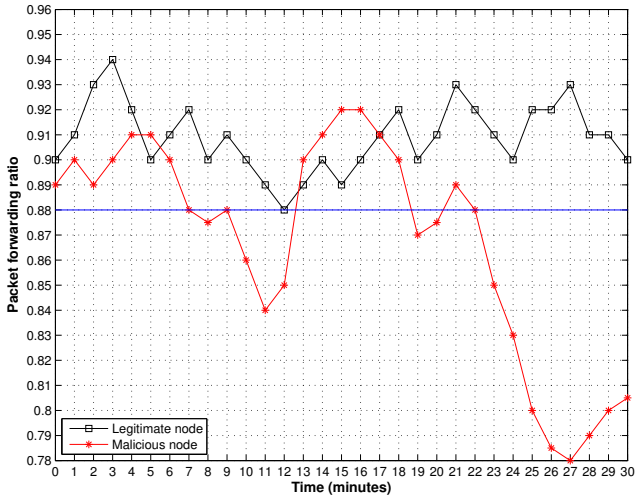


Figure 5.1: Illustration of periodic packet dropping pattern.

To illustrate the impact of these attacks on MANETs, we performed many network simulations using NS2. The simulated network is composed of 50 nodes and uses AODV as routing protocol in the presence of one malicious node that discards packets randomly over time. We monitored the behavior of nodes during a period of 30 minutes. In a first scenario, we aim to compare the behavior of that malicious node with the behavior of a normal node regarding the packet forwarding process. Figure 5.1 shows the ratios of packets forwarded by two network nodes: the first node behaves normally, while the second one performs a periodic packet dropping attack. During the 30 minutes, we noticed that the ratio of packets forwarded by the legitimate node changes between 0.88 and 0.94. On the other hand, many abnormal changes were noticed in the case of the malicious node, where the packet forwarding ratio shows a notable decrease and reaches 0.78 at $t = 27$ minutes.

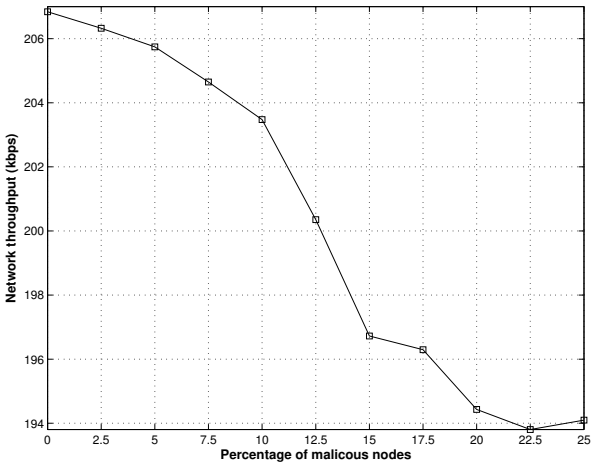


Figure 5.2: Impact of packet dropping on network throughput.

To show the impact of periodic dropping attacks on network throughput, we performed a second scenario having the same network size and simulation period that are used in the first scenario. However, we made several simulations by changing the percentage of malicious nodes in the network between 0% and 25%. According to the results obtained in figure 5.2, we notice that the network throughput decreases about 10 kbps when the percentage of malicious nodes in the network is about 15%. Based on these simulation results, we conclude that the periodic dropping attacks produce a potential perturbation of the packet forwarding process and a notable increase in network throughput. In addition, based on previous studies, we note that the malicious nodes performing this type of dropping attack may still a long period in the network without being detected [118]. Therefore, we aim to propose throughout this chapter a novel detection approach against these attacks based on a behavior analysis of network nodes and using a tracking scheme of their evolution. In the next section, we present the design of our proposed solution, and we describe the different components that we use to ensure the accurate detection of periodic dropping attacks in MANETs.

5.4 Solution overview

To address the problem presented in the previous section, we propose in this section a fully decentralized framework to recognize the future trends of network nodes based on their previous behavior evolution. Basically, this solution is designed to alleviate the uncertainty caused by the periodic dropping attacks which are often confused with legitimate packet discards caused by normal network situations.

5.4.1 System components

The main objective of our solution is to allow a node to predict the behavior of other nodes in its transmission range based on historical evaluations of their behaviors. The framework that we propose to detect packet dropping attacks consists of eight phases as shown in Figure 5.3.

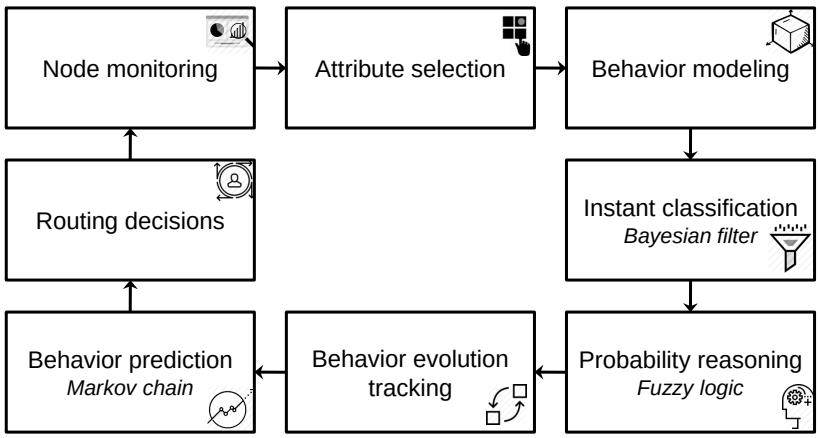


Figure 5.3: Phases of behavior prediction approach.

The input of the proposed framework is a set of information about packets exchanged by a monitored node during a defined period. The outcome is a limit probability distribution obtained through a stochastic process representing the stationary state of the monitored node. The different phases of the proposed solution are described in the following:

- Each node in the network monitors other nodes that are in its transmission range in order to collect information about packets passed by its 1-hop neighbors. Based on this information, the monitoring node models the observed behavior as a vector composed of three elements representing the forwarding ratio of three packet types: Data, RREQ and RREP. This scheme is already described in section 4.5.1.2.

- The obtained vector is used as an input of the Bernoulli classification model to evaluate the probability of maliciousness of the monitored node's behaviors during a time slot according to the scheme described in 4.5.1.3. Then, the

obtained probability value is mapped to a maliciousness level defining the behavior state of a node using a fuzzy logic model.

- Based on the sequence of maliciousness levels obtained at each time slot, the monitoring node establishes a matrix of probabilities representing the different transitions among behavior states. Then, the limit probability distribution representing the stationary state of the monitored node is predicted using a *Markov chain* model.

- Based on the obtained probability distribution, the monitoring node can determine the stationary state of the monitored node and then decide whether this node can be used to forward packets or not.

5.4.2 Detection characteristics

The aim of our proposed framework is to detect misbehaving nodes in the routing discovery and data forwarding phases. Our framework has the following security objectives:

1. Ensure a detection of different types of total, selective and periodic dropping attacks. Detecting such misbehaviors aims to maintain a reliable and secure packet forwarding among network nodes, and then guarantee the availability of routing services.

2. Propose the use of security as an attribute besides the hop count one, which is widely used as a path selection criteria in most of unsecured routing protocols such as, AODV and DSR. If this security improvement is employed to an existing routing protocol, it should not degrade its normal operations.

3. Implement a totally distributed non-cryptographic solution to ensure a detection mechanism against dropping attacks. The proposed solution should be much better than cryptographic techniques in terms of complexity and computational costs, which are not well adapted to resource-constrained networks like MANETs.

Based on these considerations, we present in the next sections the classification models and detection phases performed in the proposed framework. We prove through simulation its ability not only to detect misbehaving nodes, but also to predict their future behaviors based on historical probability evaluations.

5.5 Framework specification

This section is dedicated to describe the different techniques that we use to allow a monitoring node A to recognize the behavior of a monitored node B over a time period T composed of n time slots τ. In addition, it presents the models that we use to assign a state to a node based on its behaviors. Finally, it describes the

different phases that we use to model the transitions between states and calculate the stationary state of a monitored node.

5.5.1 Behavior-based node classification

According to the modeling scheme presented in section 4.4.2, the occurrence of a packet dropping attack is related to three elementary events regarding the forwarding of three packet types: Data, RREQ and RREP. Then, the behaviors of a node are modeled as a vector representing the forwarding ratio of these packet types.
Two schemes were described in sections 4.5.1.2 and 4.5.2.2 to model node behaviors as input vector adapted for the Bernoulli and Multinomial Bayesian filters respectively. Using these filters, we computed the probability value quantifying the likelihood that a node behaves maliciously.
According to the performance evaluation discussed in section 4.6, we prefer to adopt the Bernoulli filter to make our probabilistic classification of node's behaviors. The classification scenario is performed as follows: node A models the behavior of node B during a time slot τ as a vector composed of three boolean elements $\overrightarrow{X_\tau} = (x_{Data}, x_{RREQ}, x_{RREP})$. Then, node A calculates the probability of maliciousness of B's behavior $\overrightarrow{X_\tau}$ according to the following formula:

$$p\left(C{=}m/\overrightarrow{X}{=}\overrightarrow{X_\tau}\right) = \frac{\prod_{i=1}^{3} p\,(pt_i/m)^{x_i} \times (1-p(pt_i/m))^{(1-x_i)} \times p(m)}{\sum_{c\in\{l,m\}} \prod_{i=1}^{3} p(pt_i/c)^{x_i} \times (1-p(pt_i/c))^{(1-x_i)} \times p(C{=}c)} \quad (5.1)$$

where $\overrightarrow{X_\tau}$ is a vector of behaviors obtained during τ, pt_i is a packet type and m and l denote the malicious and legitimate classes respectively. To classify a node according to the obtained probability value, we use a fuzzy logic model instead of relying on the default classification threshold α as is defined in equation 4.11.

5.5.2 Probability reasoning model

The core idea of this process is to use two thresholds α_{min} and α_{max} instead of relying on the default classifier threshold α defined in the Naive Bayes theorem [116]. Instead of making a binary classification, namely, classifying a node as malicious or legitimate, we introduce a third class of behaviors called "suspicious". This class is used to overcome the challenge of non-detectable nodes' behaviors, and mitigate the confusion between two possible scenarios:

- Periodic dropping attack performed by a malicious node and appears as legitimate.
- Packet discarding performed by legitimate node under normal network conditions.

Hence, we consider that the suspicious class gathers those nodes' having a probability of maliciousness p belonging to a *suspiciousness interval*, denoted by I_s and represented as follows:

$$p \in [\alpha_{min}, \alpha_{max}], \ \ where \ \ \alpha_{min} = 0.5 - s \ \ and \ \ \alpha_{max} = 0.5 + s \tag{5.2}$$

where 0.5 corresponds to the value of the default classifier threshold α, and s is the range of probability values of the suspicious class. For instance, if I_s is set to 0.05, then the interval $[\alpha_{min}, \alpha_{max}]$ is set to $[0.45; 0.55]$.

Since we decided to extend the binary classification by introducing a third class, we should define a reasoning model of the obtained probability value and we prefer to use a fuzzy logic model. This mode is derived from fuzzy set theory dealing with reasoning that is "approximate" rather than "precisely" deduced from classical predicate logic [120]. This theory is described as a mathematical system based on a *membership function* that uses *truth value* between 0 and 1 as input, and has a *linguistic variable* as output. In our case, the only truth value is the probability of maliciousness p, and the linguistic variables are defined by three maliciousness levels: *legitimate*, *suspicious* and *malicious*.
There are many types membership functions defined in fuzzy logic theory that can be used according to several criteria [121]. These functions are out the scope of this thesis, however, we use the graphical representation of the Gaussian membership function to illustrate our fuzzy logic model. Figure 5.4 represents the membership between the probability of maliciousness values and behavior classes. We note that Gaussian function is used only to clarify our idea graphically, and it is not used further in our reasoning model.

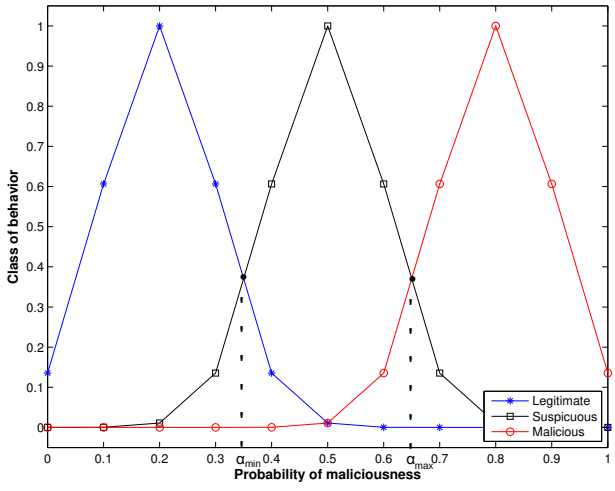


Figure 5.4: Membership function between probability values and behavior classes.

In this figure, the interval of probability values of suspicious class is set to $[0.35; 0.65]$. We note that the *full membership* is represented by 1, and *no membership* by 0. Then, the full membership to the legitimate, suspicious and malicious

classes is respectively mapped to 0.2, 0.5 and 0.8 probability values.
Fuzzy logic usually uses IF/ THEN rules, which are usually expressed in the form: IF *variable* IS *set* THEN *action*. In our fuzzy logic model, the *fuzzification* process is based on 7 *propositions* and 3 *rules* detailed in the following.

Propositions

pr_1: the value of p is less than α_{min}.
pr_2: the value of p is greater than α_{min}.
pr_3: the value of p is less than α_{max}.
pr_4: the value of p is greater than α_{max}.
pr_5: a legitimate behavior is detected.
pr_6: a suspicious behavior is detected.
pr_7: a malicious behavior is detected.

Fuzzy rules

R_1: if pr_1 THEN pr_5.
R_2: if pr_2 AND pr_3 THEN pr_6.
R_3: if pr_4 THEN pr_7.

The fuzzy logic model presented above is an essential phase to find out a solution that allows us to distinguish "*legitimate nodes* dropping packets due to normal network conditions" and "*malicious nodes* performing a periodic packet dropping attack". We believe that the tracking of behavior's evolution of a node can help to mitigate the uncertainty level induced by such situations. Therefore, we propose in the next subsection a Markov chain model to identify the *stationary state* of a node based on its previous evolution.

5.5.3 Stochastic modeling of node's evolution

Based on the fuzzy logic model presented in the previous subsection, we use the defined maliciousness levels to represent a space of possible states of a network node: $E = \{l, s, m\}$. We denote with X_t and $X_{k,t}$ respectively the state of the network and the state of node k at a time t. $(X_t)_{t \in \mathbb{N}}$ is the stochastic process modeling the network evolution, such that:

$$X_t = \bigcup_{k=1}^{N_n} X_{k,t}$$

where N_n represents the number of nodes in the network. We represent the evolution of the network as a Markov chain, where the process $(X_{k,t})_t$ is memory-less and that conditionally at the present, the past and the future are independent for all k, which can be expressed mathematically as following:

$$p\left(X_{k,t+1}{=}x_{k,t} / X_{k,t}{=}x_{k,t},\, X_{k,t-1}{=}x_{k,t-1},\, \ldots,\, X_{k,0}{=}x_{k,0}\right) = p\left(X_{k,t+1}{=}x_{k,t} / X_{k,t}{=}x_{k,t}\right) \tag{5.3}$$

In our case, the memory-less property means that the state of a node at future time $t+\tau$ relies on the node state at the current time t and does not depend on the state at earlier time instants $t, ..., t-\tau$.

Let $p_{k,ij}$ the "*probability of transition of a node k*" from state i to state j expressed by $p_{k,ij} = p(X_{k,t+1} = j/X_{k,t} = i)$. In our case, we assume that the evolution of the network is homogeneous, namely, the probability of transition from a state to another is constant over time, which can be written as follows:

$$p_{k,ij} = p\left(X_{k,t+1}=j/X_{k,t}=i\right) = p\left(X_{k,1}=j/X_{k,0}=i\right),\ \forall\ (i,j) \in E^2, and\ \ t \in \mathbb{N} \quad (5.4)$$

5.5.3.1 Transition matrix establishment

After defining the states of our Markovian process and their different properties, we use a Markovian stochastic matrix denoted by $(X_{k,t})_t$ to represent the evolution of node over T. This matrix contains the probabilities of transitions between different behavior states of a node k, which is denoted by $P^k = (p_{k,ij})_{i,j\in E}$ and represented as follows:

$$P_k = \begin{pmatrix} p_{k,ll} & p_{k,ls} & p_{k,lm} \\ p_{k,sl} & p_{k,ss} & p_{k,sm} \\ p_{k,ml} & p_{k,ms} & p_{k,mm} \end{pmatrix} \quad (5.5)$$

where each entry $p_{k,ij}$ denotes the conditional probability that node k moves from present state i to next state j. For instance, $p_{k,ls}$ designates the probability that a node k is suspicious given legitimate. In our case, the probability value of each matrix entry is obtained using the following formula:

$$p_{k,ij} = p_k\left(X_{t+1} = j \mid X_t = i\right) = \frac{\sum Tr_{(i,j)}}{\sum C_i} \quad (5.6)$$

where $Tr_{(i,j)}$ is the number of times that node k has moved from present state i to next state j, and C_i denotes the number of times that k has visited state i.

5.5.3.2 Limit probability distribution

The intrinsic purpose of using a Markov chain model in our scheme consists of predicting the stationary state of a node based on its evolution. We denote with $\nu_{k,i}(t)$ the probability that node k being in state i at time t, namely, $\nu_{k,i}(t) = p\{X_{k,t} = i\}$. At any time t, we obtain the stochastic vector $\nu_k(t)$, such that $\nu_k(t)$ verifies the following matrix for $t \in \mathbb{N}$:

$$(\nu_{k,l}(t+1), \nu_{k,s}(t+1), \nu_{k,m}(t+1)) = (\nu_{k,l}(t), \nu_{k,s}(t), \nu_{k,m}(t)) \times \begin{pmatrix} p_{k,ll} & p_{k,ls} & p_{k,lm} \\ p_{k,sl} & p_{k,ss} & p_{k,sm} \\ p_{k,ml} & p_{k,ms} & p_{k,mm} \end{pmatrix} \quad (5.7)$$

which can be also written as $\nu_k(t+1) = \nu_k(t) \times P_k, t \in \mathbb{N}$. Therefore, the Markovian chain representing the evolution of node k is entirely characterized by its transition

matrix P_k and initial transition probability $\nu_k(0)$. Noting that P_k^n is the n^{th} power of the matrix P_k, we have:

- $\nu_k(1) = \nu_k(0) \times P_k$
- $\nu_k(2) = \nu_k(1) \times P_k = \nu_k(0) \times P_k^2$
- $\nu_k(3) = \nu_k(2) \times P_k = \nu^k(0) \times P_k^3$

  .
  .
  .
- $\nu_k(n) = \nu_k(0) \times P_k^n$ (5.8)

According to [122], a finite Markov chain has at least one stationary probability distribution. In our case, the Markov chain representing the evolution of a node has a space of three possible states. Therefore, the transition matrix P_k has at least one stationary probability distribution denoted by $\lim_{n\to\infty} \nu_k(n)$, and then the Markov chain representing the evolution of a node admits a stationary state.
On the other hand, based on the ergodicity theory presented in [123], there is at least a power of transition matrix for any network node having strictly positive elements, and then our Markov chain is ergodic. In fact, an ergodic Markov chain defined with its stochastic vector $(\nu(n))_{n\in\mathbb{N}}$ has a unique limit distribution $\lim_{n\to\infty} \nu(n)$ which does not depend on its initial vector $\nu(0)$. Therefore, in our case, any node k has a unique stationary distribution which is given by $\lim_{n\to\infty} \nu_k(n)$ and consequently by $\lim_{n\to\infty} P_k^n$.
Based on the stochastic process described in this section, we present in the next subsection the whole prediction process which can be performed to ensure the detection of periodic packet dropping attacks in MANETs.

5.5.4 Behavior-based prediction algorithm

As we already mentioned, the scope of the detection approach presented in this chapter is to ensure a reliable and secure routing decisions by verifying the state of network nodes based on their behaviors.

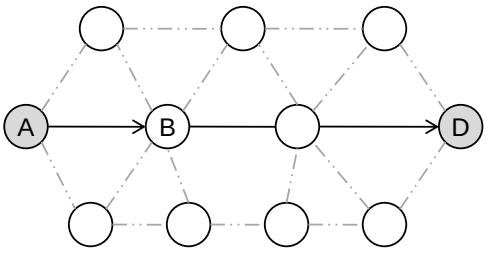


Figure 5.5: Verification of neighbor's behavior.

In figure 5.5, we suppose that node A has diffused a route request in the network

in order to find out a route to a node D using a reactive routing protocol such as, AODV. Being a non-secured protocol, AODV returns to A a route composed of the minimal number of intermediate nodes, in order to deliver packets to D as fast as possible, without considering the possibility that the found route may include malicious nodes.

Algorithm 1 *Computing the limit probability distribution of node B*

{***INPUT*** → ***Node*** A ***tracks the behaviors of a node*** B ***during*** T}
{***Collecting information about packet forwarding***}
while $(t < T)$ **do**
  $X[3] \leftarrow (x_{RREQ}, x_{RREP}, x_{DATA})$; {*Behavior's vector*}
  $P[t] \leftarrow P(Malicious/\ X)$; {*Probability of maliciousness*}
  if $(P[t] < \alpha_{min})$ **then**
    $C[t] \leftarrow l$; {*Legitimate behavior*}
  else if $(P[t] \geq \alpha_{min}$ && $P[t] < \alpha_{max})$ **then**
    $C[t] \leftarrow s$; {*Suspicious behavior*}
  else
    $C[t] \leftarrow m$; {*Malicious behavior*}
  $t \leftarrow t + \tau$; {*Next time slot*}
{***Counting occurrences of transitions in each class*** C}
for $(i = 1;\ i \leq 3;\ i \leftarrow i + 1)$ **do**
  for $(j = 1;\ j \leq 3;\ j \leftarrow j + 1)$ **do**
    $P[i][j] \leftarrow Count(C,\ i,\ j)/Count(C,\ i)$; {*Transition matrix establishment*}
{***Obtaining the limit distribution***}
$step = 2$;
while $(P[1][1] \neq P[2][1]$ || $P[1][1] \neq P[3][1])$ **do**
  $step \leftarrow step + 1$; {*Increment power*}
  $P \leftarrow P^{step}$; {*Calculate the powers of transition matrix*}
{***OUTPUT*** → ***Limit probability distribution of node*** B}

By implementing our solution, A should consider the reliability of its next hop as an additional criteria when selecting routes before forwarding packets. Therefore, A should perform a verification of B's state according to algorithm 1, in order to decide whether B is reliable to forward packets or not.
Based on previous observations, node A affects a behavior state of node B at each time slot τ over a time period T. Then, A represents the sequence of instant affectations made to B's behaviors over T as a transition matrix according to equation 5.7. Finally, based on the ergodicity property of the obtained Markov chain, A is able to obtain a unique limit probability distribution representing the stationary state of B.

5.5.5 Routing decision process

After obtaining the stationary state of node B, we introduce a decision process which enables node A to select (respectively avoid) node B to route packets (respectively from packet forwarding) based on its limit probability distribution.

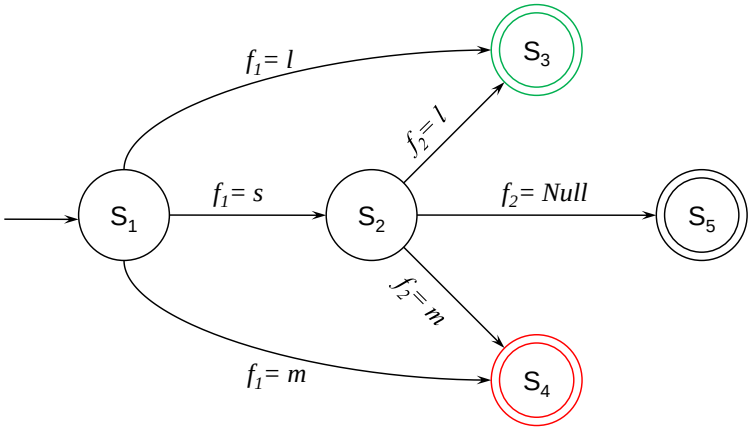


Figure 5.6: FSM model for reliable routing decision algorithm.

Figure 5.6 shows a Finite State Machine (FSM) describing the decision process that we propose to use before forwarding packets through node B. The decision is based on two functions and five states as follows:

- f_1: an initial comparison function to make an intermediate decision based on $MAX\,(P(X_B{=}l), P(X_B{=}s), P(X_B{=}m))$.
- f_2: an intermediate comparison function to make a final decision based on $MAX\,(P(X_B{=}l), P(X_B{=}m))$.
- S_1: the initial state applying the f_1 function.
- S_2: an intermediate state applying the f_2 function.
- S_3: a final state representing a positive decision about node B.
- S_4: a final state representing a negative decision about node B.
- S_5: a final state representing a fail in making decision about node B.

The FSM model takes a vector of probability distribution as input. If it finds that the maximal probability value is $P(X_B{=}l)$ (respectively $P(X_B{=}m)$), then it considers B as legitimate (respectively malicious). However, if it finds that $P(X_B{=}s)$ has the maximum value after the first comparison, it makes a second comparison between $P(X_B{=}l)$ and $P(X_B{=}m)$ to make a decision as follows:

- If it finds that $P(X_B = l)$ is greater than $P(X_B = m)$, it deduces that the high level of uncertainty is rather caused by packet discarding under normal network situations.
- If it finds that $P(X_B = m)$ is greater than $P(X_B = l)$ it deduces that the high level of uncertainty is rather caused by a periodic packet dropping attack.

On the other hand, if the FSM reaches the final state S_5, there are two possible cases:

- The probabilities of transition $P(X_B = l)$, $P(X_B = s)$ and $P(X_B = m)$ are equal.
- The limit distribution of B has $P(X_B = s) = 1$.

In both cases, we deduce that it is not possible to make a decision based on the limit distribution of node B, and then our solution cannot determine the stationary state of B.

5.6 Performance Evaluation

Throughout this chapter, we presented a prediction framework that combines a probabilistic classification, fuzzy logic and a Markov chain model. The rationale behind this extensive study of node's behaviors is to detect malicious nodes that perform a packet dropping attack which is closely similar to normal dropping events. Basically, our solution is based on the following hypothesis: "to what extent can the node evolution tracking improve the detection of malicious nodes, especially those having an uncertain behavior?".
To validate this hypothesis, we first prove through network simulations the ability of our solution to predict the stationary state of a node based on its previous behaviors evolution.
Secondly, to overcome the confusion between malicious and legitimate packet discards, we introduced the notion of suspicious level of node's behavior. We prove through simulation the importance of introducing this level to ensure the detection of periodic dropping attacks with a high rate of accuracy.

5.6.1 Validation of ergodic theory

We already showed in section 5.5.3.2 that the network evolution can be represented by an ergodic Markov chain, and then any node in the network admits a unique stationary state. To prove the validity of this hypothesis, it is sufficient to demonstrate for one network node a transition matrix having only strictly positive elements. Then, we have to prove that this node has a stationary state represented by a unique limit probability distribution.
To validate this hypothesis, we apply in the following the prediction process presented in algorithm 1 on a simulated network illustrating the scenario described in

figure 5.5. The simulation scenario is elaborated during a time period of 30 minutes. We declare node A as a source node having packets to send to another node D through its 1-hop neighbor B. We assume that A wants to evaluate the legitimacy of B based on its evolution during a monitoring period equal to T. We set node B to malicious mode, where packets passing through it are randomly discarded over T.

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
state	•	−	+	+	−	−	−	+	−	−	−	−	−	•	•
t	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
state	−	−	•	+	−	−	−	−	+	•	−	•	+	−	−

Transition (i→j)	Occurrences	p (X^B=j/ X^B=i)
+ → +	1	1/6
+ → •	1	1/6
+ → −	4	2/3
• → +	2	1/3
• → •	1	1/6
• → −	3	1/2
− → +	3	3/17
− → •	3	3/17
− → −	11	11/17

$$\begin{pmatrix} 0.166 & 0.166 & 0.668 \\ 0.166 & 0.333 & 0.500 \\ 0.176 & 0.176 & 0.648 \end{pmatrix}$$

Figure 5.7: Process of evolution tracking of node B performed by node A during T. (+): legitimate, (−): malicious, (•): suspicious

Figure 5.7 illustrates the steps performed by A to represent the evolution of node B. First, A selects a time slot equal to $\tau = 1$ minute to compute the probability of maliciousness of B. Then, it affects for each observation a state defining the behaviors of B observed during τ according to the suspiciousness interval configuration I_s. This latter is set to 0.2, which means that $[\alpha_{max}, \alpha_{max}] = [0.4, 0.6]$.
First, A obtains a sequence of 30 states representing the evolution of B during the 30 minutes. Then, node A computes the probability for each occurred state transition according to equation 5.6. For instance, the value of the entry at the 1^{st} line and the 3^{rd} column of the matrix represents $p_{B,lm}$, and is obtained as follows:

$$p_{B,lm} = \frac{\#\ of\ times\ that\ B\ has\ moved\ from\ l\ to\ m}{\#\ of\ times\ that\ B\ has\ visited\ the\ state\ l}$$

By performing this calculation for different possible transitions, A deduces the Markovian matrix of B denoted by p_B^1 as following :

$$P_B^1 = \begin{pmatrix} 0.166 & 0.166 & 0.668 \\ 0.166 & 0.333 & 0.500 \\ 0.176 & 0.176 & 0.648 \end{pmatrix}$$

which can be also illustrated using a directed graph as depicted in figure 5.8, where the nodes are the possible node states of $E = \{l, s, m\}$, and the edges are the probabilities of transitions between these states.

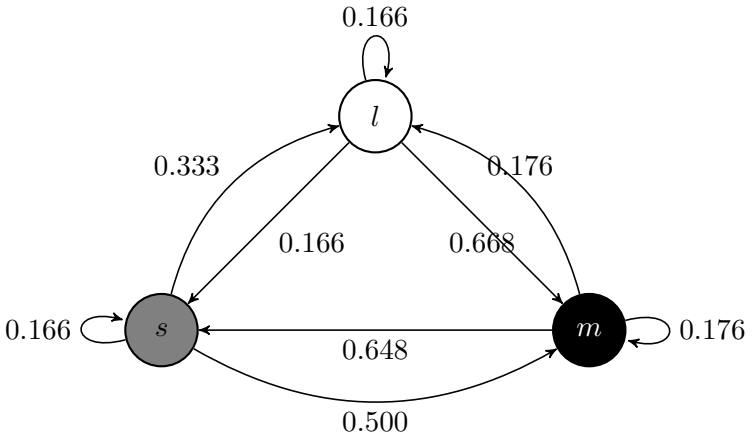


Figure 5.8: Directed graph representing the transition matrix P^B.

To predict the limit probability distribution of node B, node A computes the $\lim_{n\to\infty} P_B^n$ by performing the powers of the initial transition matrix P_B^1 according to equation (5.8). For instance, the 2^{nd} and 3^{rd} powers of P_B^1 are denoted respectively by P_B^2 and P_B^3 an obtained as follows:

$$P_B^2 = \begin{pmatrix} 0.2007 & 0.1728 & 0.6264 \\ 0.1989 & 0.1713 & 0.6298 \\ 0.2020 & 0.1727 & 0.6253 \end{pmatrix}, \; P_B^3 = \begin{pmatrix} 0.2013 & 0.1725 & 0.6262 \\ 0.2011 & 0.1725 & 0.6264 \\ 0.2014 & 0.1725 & 0.6263 \end{pmatrix}$$

The power n is incremented until the lines of transition matrix become identical, which indicates that the stationary distribution is reached after a number of steps equal to n. Therefore, A finds that the stationary distribution of B is reached at the 4^{th} step, where the transition matrix P_B^4 is equal to:

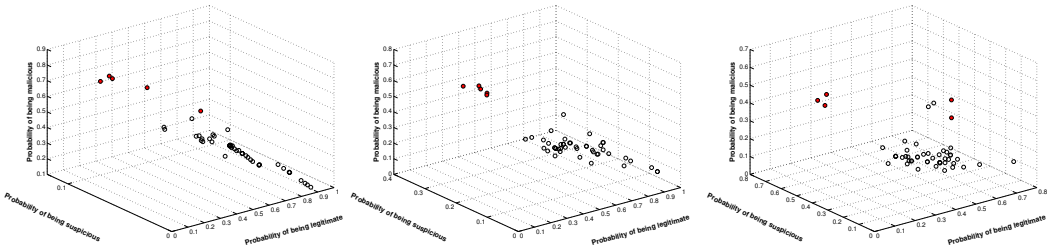
$$P_B^4 = \begin{pmatrix} 0.2012 & 0.1725 & 0.6263 \\ 0.2012 & 0.1725 & 0.6263 \\ 0.2012 & 0.1725 & 0.6263 \end{pmatrix}$$

Consequently, the stationary distribution has the following probability values: $P(X_B = l) = 0.2012$, $P(X_B = s) = 0.1725$ and $P(X_B = m) = 0.6263$. According to the FSM model presented in figure 5.6, A deduces that B is a malicious node

since its probability of maliciousness $P(X_B = m)$ is greater than its probability of legitimacy $P(X_B = l)$ and suspiciousness $P(X_B = s)$. Therefore, A decides to avoid B when forwarding packets.

5.6.2 Study of uncertainty threshold

As we already mentioned, we introduce the suspicious level as an attempt to overcome the confusion between legitimate and malicious packet discarding. We showed that a node is classified as suspicious if its probability of maliciousness belongs to an interval centered on 0.5 and having α_{min} and α_{max} as lower and upper bounds respectively. We denoted with I_s the suspiciousness interval defined by $\alpha_{max} - \alpha_{min}$. I_s is a critical parameter that we aim to adjust in order to obtain the most reliable routing decisions.



(a) $I_s = 0.1$ (b) $I_s = 0.2$ (c) $I_s = 0.3$

Figure 5.9: Limit probability distribution of network nodes.

In figure 5.9, we evaluate the impact of I_s interval on the limit probability distribution in the case of a network composed of 50 nodes, where 10% of them are malicious. The black dots represent the distribution of legitimate nodes, while red filled ones represent the malicious nodes distribution.
According to figure 5.9a, when we set the value of I_s to 0.1, the scatter shows an unclear relationship between the points of the class due to a notable variation between their probability distributions.
However, in figure 5.9b, we note that the points of the same class are mostly grouped into a clear linear shape when I_s is set to 0.2. The legitimate nodes have high values of probability of legitimacy, while the malicious nodes have high values of probability of maliciousness, with a low value of uncertainty in both cases. Therefore, the classification of nodes shows a better performance than that in the case of $I_s = 0.1$.
Finally, when we increase the value of I_s to 0.3, we note a significant variation between the points representing the class of malicious nodes. In addition, there are high values of probability of suspiciousness, which may lead further to a confusion between malicious and legitimate nodes.
Consequently, using this data set of network nodes we note that $I_s = 0.2$ is the most appropriate threshold configuration since it ensures the best classification of nodes.
In the following subsection, we elaborate several simulations using network scales

less than 50 nodes. We set the value I_s to 0.2 in all simulations in order to evaluate the performance of our solution in function of monitoring time and percentage of malicious nodes.

5.6.3 Evaluation of detection accuracy

To evaluate the performance of our proposed solution we performed many simulations using NS2 simulator by changing three main parameters: network size, percentage of malicious nodes and simulation time. The simulated network is composed of nodes having a low mobility speed between 2 and 8 m/s and a transmission range of 250m. In each simulation scenario, 20% of network nodes establish communication sessions and exchange data packets using a *Constant Bit Ratio* (CBR) traffic. As we already mentioned, the suspiciousness interval I_s that we use to assign a state to a node is set to 0.2 in all simulations. The rest of parameter settings of our experimental setup are depicted in table 5.1.

Table 5.1: Network simulation parameters.

Parameter	**Value**
Coverage area	1000m × 1000m
Number of nodes	10, 20 and 50 nodes
Transmission range	250m
Simulation time	10, 20 and 30 minutes
Time slot τ	1 minute
Mobility model	Random Waypoint
Min/Max speed	$[2-8]$ m/s
Routing protocol	AODV
Traffic type	UDP/ CBR
% of malicious nodes	10, 30 and 50%
I_s interval $(\alpha_{max} - \alpha_{min})$	0.2
MAC layer type	IEEE 802.11p

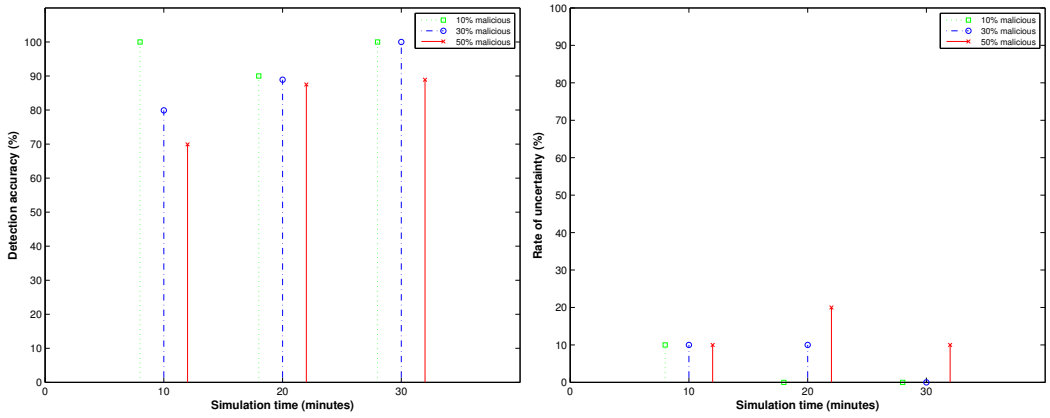
The simulated network contains a changing number of malicious nodes which runs an instance of periodic packet dropping algorithm that we elaborated in C^{++}. The data sets of our testbed are the trace files collected at the end of each simulation scenario. These files are processed using *AWK* and *Python* scripts, in order to extract information and perform a behavior modeling based on selected attributes. The probability calculation process is implemented in C^{++}, and used as input for transition matrices establishment which is performed in *MATLAB* environment. In addition, we evaluate the rate of nodes that are not detected by our solution due to their uncertain state. Hence, the performance of our solution is evaluated based on the detection accuracy (Acc) and uncertainty (Unc) such that:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad and \quad Unc = \frac{\#\ non - detected\ nodes}{Total\ number\ of\ nodes}$$

where TP, TN, FP and FN denote respectively the rate of *True Positive*, *True Negative*, *False Positive* and *False Negative* as they were already explained in section 4.6.4. To facilitate the interpretation of the results that we obtained in different simulation scenarios, we use sometimes the following notations:

- $\%_{unc}$: the rate of non-detected nodes.
- T_{sim}: the duration of a simulation scenario.
- $\%_{mal}$: the percentage of malicious nodes in the network.

Figure 5.10a shows the detection accuracy of our proposed solution when the network is composed of 10 nodes. We note a full accurate detection when $\%_{mal}$ is equal to 10%, except when the simulation time is set to 20 minutes, where the detection accuracy is equal to 90%. When $\%_{mal}$ is set to 30%, we note a lower detection accuracy equal to 80% when T_{sim} is set to 10 minutes. However, our solution provides a full accurate detection of malicious nodes when we increase T_{sim} to 30 minutes.
On the other hand, figure 5.10b shows the rate of nodes which are not recognized at all by our solution. When the percentage of malicious nodes is set to 10%, we note that 10% of nodes are not identified when T_{sim} is set to 10 minutes. However, this uncertainty disappears once T_{sim} is greater than 20 minutes. Similarly, in the second scenario, when $\%_{mal}$ is increased to 30%, the behaviors of 10% of nodes are not recognized unless we increase T_{sim} to 30 minutes. Finally, when $\%_{mal}$ is 50%, the percentage of non-detected nodes becomes greater than 10% whatever the simulation time, and it reaches a rate of 20% when T_{sim} is equal to 20 minutes.

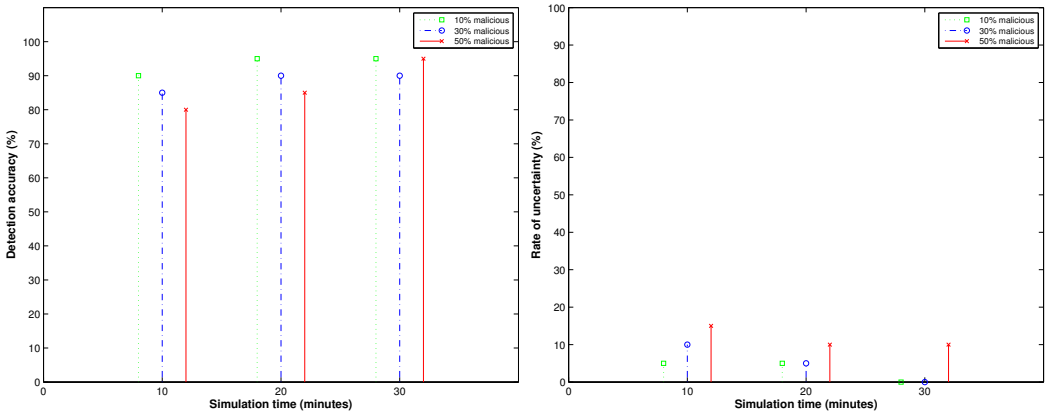


(a) Detection accuracy vs simulation time (b) Rate of uncertainty vs simulation time

Figure 5.10: Network size = 10 nodes

In Figure 5.11a we evaluate the detection accuracy of our solution when the network size is equal to 20 nodes. When we set $\%_{mal}$ to 10%, the accuracy rate remains between 90% and 95% regardless the duration of simulation scenario. Similarly, when the $\%_{mal}$ is increased to 30% we get a lower accuracy rate that persists between

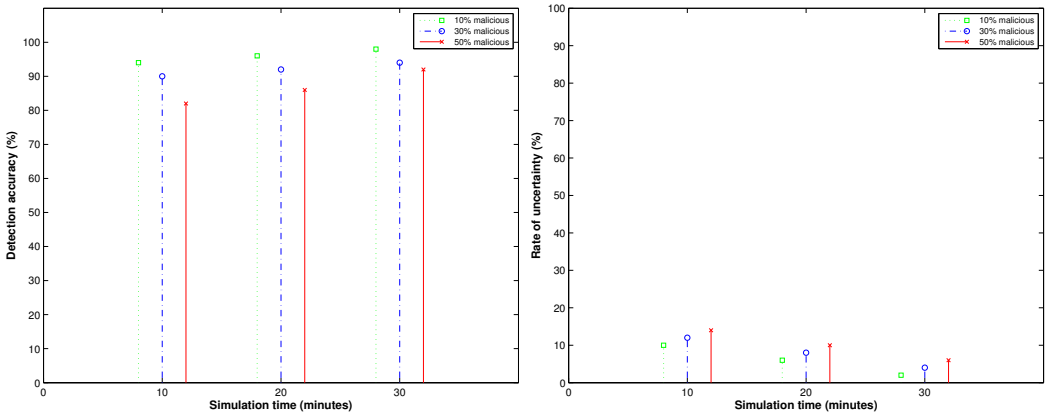
85% and 90% in the three configurations of T_{sim}. However, in the case of $\%_{mal}$ is equal to 50%, the detection accuracy increases from 80% to 95% in the three cases. Regarding the rate of uncertainty, figure 5.11b shows that a percentage of 5% of nodes are not detected when the value of $\%_{mal}$ is equal to 10%, and this percentage disappears when we increase T_{sim} to 30 minutes. In the case of 30 malicious nodes, we note a value of $\%_{unc}$ equal to 10% when the T_{sim} is set to 10 minutes which appears progressively when T_{sim} is increased to 30 minutes. Finally, when $\%_{mal}$ is increased to 50% we note that there are always nodes that cannot be identified with a $\%_{unc}$ value between 10% and 15% whatever the simulation duration.



(a) Detection accuracy vs simulation time (b) Rate of uncertainty vs simulation time

Figure 5.11: Network size = 20 nodes

The last configuration of network size that we use in our simulations is set to 50 nodes, where the results are depicted in figure 5.12. According to figure 5.12a, when we set $\%_{mal}$ to 10% the detection rate increases to 90%.



(a) Detection accuracy vs simulation time (b) Rate of uncertainty vs simulation time

Figure 5.12: Network size = 50 nodes

However, when we increase $\%_{mal}$ to 30% we note that the accuracy rate decreases

to 85% in the case of $T_{sim} = 10$ minutes. By maintaining the same percentage of malicious nodes, we increase the simulation time to 30% and we note that the accuracy increases slightly and reaches a rate of 90%. When $\%_{mal}$ is set to 50% the accuracy rate is equal to 80% when T_{sim} is set to 10 minutes, and has a significant increase that reaches a rate of 95% when we increase the T_{sim} to 95%.

5.7 Discussion

Based on the performance evaluation presented in the previous section, we underline some global interpretation regarding the impact of network configuration on the detection accuracy.

- The detection is more accurate in the case of large scale networks (50 nodes) than that of small ones (10 nodes), where the malicious nodes can be detected with an accuracy rate greater than 90% whatever their percentage in the network. In addition, in the case of large scale networks, the rate of non-detected nodes does not exceed 15% whatever the proportion of malicious nodes in the network.

- Regarding the simulation time, more the monitoring duration is long the more the detection is accurate whatever the network size. On the other hand, the rate of non-detected nodes decreases when the simulation time increases. Therefore, the detection of periodic dropping attacks is more accurate when the monitoring phase is performed during a long time period.

- The malicious nodes that we attempt to detect in the above simulation scenarios perform a periodic packet discarding, in such way they appear as closely similar to a legitimate packet discarding. Despite all these technical challenges that we attempted to simulate, the results show an average value of accuracy equal to 90% in different network configurations. Thus, we believe that an adequate tracking of node's behaviors changes is essential to ensure the detection of periodic dropping attacks.

- The performance of the behavior monitoring process in the case of highly mobile networks is not handled in this work. We assumed that the simulated networks are composed of nodes having a low mobility speed. However, to address this problem, we aim to realize in our future works more investigation of the impact of the mobility speed on the choice of T and τ values which are the key parameters of our monitoring process.

5.8 Conclusion

In this chapter, we presented a decentralized framework based on a behavior prediction model to detect periodic dropping attacks in MANETs. We firstly showed the difficulty of distinguishing these attacks from the normal packet discards and

their potential impact on network performance. We presented the general design of different phases composing our framework and we described the interaction between them.
Regarding the monitoring scheme, we used the same scheme proposed in chapter 4 to collect information about exchanged packet types, model the collected behavior and calculate the probability of maliciousness. Then, we described the fuzzy logic model that we introduced to assign a level of maliciousness of a node based on the obtained probability value. We defined a space of possible states of a node's behavior to represent the transition between states using Markov chain model. We showed through simulation on NS2 and MATLAB the ability to predict the stationary state of a node based on its previous evolution. Finally, the obtained simulation results shows that our solution is able to detect periodic dropping attacks with an accuracy rate greater than 90%.

Chapter 6

Conclusion and future directions

In this chapter, we conclude the outcome of our research work. We first overview the solutions that we proposed to detect DoS attacks at the ad hoc routing layer. Then, we give some technical challenges that we deduced following our global evaluation of the proposed solutions. Finally, we overview other research contributions that we realized during the thesis and we give some directions of our future work.

6.1 Conclusion

In this thesis, we have proposed two decentralized approaches to detect packet dropping attacks in MANETs based on a behavior analysis of network nodes.
We first presented an overview of security attacks at the ad hoc routing layer, precisely those leading to a DoS. We classified DoS attacks in three broad categories that we consider as the most potential threats against routing availability: packet dropping, resource consumption and routing disruption. We inquired their techniques, objectives and damaging effects on routing performance. Based on previous studies and our own network simulations, we deduced that packet dropping attacks constitute a severe threat against routing services availability which needs more investigation. We surveyed and discussed certain mechanisms and countermeasures that are proposed in the literature to secure the ad hoc routing protocols against DoS attacks. We focused on solutions that are proposed to detect packet dropping attacks, especially those based on classification models, and we introduced the original features of our solution in comparison with these solutions.
We then proposed a behavior-based detection mechanism using the Bernoulli and Multinomial Bayesian classification models. We described the decentralized operations of the detection phases starting with a statistical analysis of packets through a node's transmission range. We selected the attributes that we consider as the most representative to model the behavior of network nodes. We described the steps of the probabilistic classification models of behavior models and evaluated their performance using different threshold configurations. Simulation results showed an

efficient detection of packet dropping attacks using the Bayesian classification models. A full detection of dropping attacks can be ensured by combining the advantages of the two models.
We also exploited the aforementioned classification framework to propose a prediction framework of node's behaviors using a stochastic process. We proved that the detection of periodic dropping attackers can be improved by tracking the evolution of their behaviors. We defined the space of possible states of node's behaviors based on a fuzzy logic model and represented the transitions among these states using a Markov chain model. We showed through network simulations the ability of our solution to predict the state of a node based on its previous evolution, and its efficiency to detect periodic packet dropping attacks with an accuracy rate greater than 90%.

6.2 Emergence of new technical challenges

The solutions proposed in this thesis aim to address the problem of routing services availability in MANETs. The detection and prediction mechanisms are basically complementary and designed to ensure a full detection of different types of packet dropping attacks. However, there are some technical challenges that we deduced following our global evaluation regarding the attributes selection for behavior modeling and nodes' classification.

In the probabilistic classification model that we proposed, we used the forwarding rates of data, RREQ and RREP packets as classification attributes. This selection is based on the fact that the behaviors of a node can be represented by evaluating its participation in data forwarding, routing discovery and routing establishment services. We intentionally omitted the possible manipulation of RERR and Hello packets in the case of AODV routing protocol, by assuming that their dropping has a negligible damage on routing availability in comparison with the packet types that we selected. After further studies, we found that disrupting the maintenance of established routes, which is ensured by RERR and Hello messages, may also downgrade the routing services.
Exchanging Hello packets is used to maintain connections between those nodes that are in the transmission range of each other. Since these packets are 1-hop RREP packets, they are usually consumed by the receiving node after having informed about a neighbor's presence without any further use. Hence, discarding these packets by a node does not have any effect on routing performance. However, altering Hello packets may be exploited by a malicious entity to divert the traffic from its normal destination, which is actually out the scope of this thesis.
On the other hand, the RERR packets are used to inform network nodes about the occurrence of a link failure. This packet is sent by the node which notices the failure information towards nodes that use that link to send their packets. If a malicious node discards a RERR packet that are supposed to be forwarded,

then the network nodes which are assumed to be informed about the link failure may use a broken link to forward their packets. Therefore, the packets passing through that link are susceptible to be lost. Consequently, RERR dropping can disrupt the network performance and threat the performance of routing services.

In both solutions presented in chapter 4 and 5, the monitoring node relies only on its direct observation to decide whether another monitored node is malicious or not. However, in most of trust-based detection mechanisms, the decision about a node's behavior is made based on the aggregation of direct observations and the recommendations of other nodes. In this case, nothing prevents a malicious node to give bad recommendations about other nodes to disrupt the trust evaluation phase [48]. Therefore, our solutions are less vulnerable to this kind of attacks since the observations about nodes are made without any aggregation with other nodes' recommendations.
However, this way of decision making may lead to false alerts when detecting malicious nodes especially in dense networks. For instance, due to a network congestion a bottleneck node may be considered as malicious due to the high rate of packets it discards. However, the aggregation of direct and indirect observations can mitigate such faulty decisions, since a node that is detected as malicious by another node may be classified as legitimate by other nodes. Therefore, using the recommendation of other nodes as a second hand information may help to overcome the challenge of false alerts in such situations.

Proposing a resource-aware detection solution was not a priority in this thesis. The complexity of monitoring operations, behavior modeling and classification algorithms was neither treated nor evaluated in our research works. Moreover, the memory space and the computational overhead that required to perform the evolution tracking process proposed in chapter 4 was not handled. Since the availability of routing services is the most important objective of our research works, we plan to more focus on resource constraints of network nodes.

6.3 Research works in progress

To address the problem of routing availability threats, we realized two research contributions that treat the flooding and tunneling attacks. According to DoS attacks taxonomy proposed in chapter 2, these attacks may lead respectively to a network resource exhausting and perturbation of routes in MANETs. We investigated the techniques used by these attacks and interpreted their damaging effects on routing services availability.
We proposed a preliminary solution against tunneling attacks in MANETs based on a graph theoretical model, which addresses precisely the encapsulation-based Wormhole technique. The solution that we presented is designed to be used by a node before sending a RREP packet as response to a received RREQ packet. It

consists of verifying whether a received RREQ is diffused through a malicious node by comparing its hop count with that of other nodes that have diffused the same RREQ. This contribution is published as a conference article; more details about this solution can be found in [49].
On the other hand, we investigated the RREQ flooding attacks that can target the ad hoc routing and exhaust the resources of network nodes. We designed a solution to enable a node to detect the occurrence of such attacks based on a statistical approach. The core idea of this approach is to monitor and evaluate the *Exponential Weighted Moving Average* (EWMA) of RREQ packets generated by a node, in order to detect the abnormal route request generations. This work was published in the proceedings of a international conference in [50].

6.4 Future research directions

The work realized in this thesis motivates us to elaborate the following short-term research works:

- **Adaptability of solutions**: both solutions proposed in chapter 4 and 5 deserve more studies in terms of **detection accuracy**. A further analysis is necessary of the **monitoring phase** regarding the hardware capability of network nodes to work in promiscuous mode. On the other hand, we need to refine theoretically the detection criteria of dropping attacks, by studying the impact of **threshold** configurations on the detection accuracy. Further simulations can be also carried out to evaluate the performance of detection phases in **large scale** networks.

- **Node mobility**: the prediction framework presented in chapter 5 needs more investigation regarding the process of behavior evolution tracking. The long time monitoring process should be studied, especially in **highly mobile ad hoc networks**, where the behavior tracking is challenging.

- **Detection uncertainty**: according to simulation results obtained in chapter 5, there is always a percentage of nodes ($< 5\%$) in the network that cannot be identified using our solution due to their uncertain behaviors. We need to make further studies of the **impact of thresholds configurations** and **monitoring time** on the percentage of non-detected nodes.

- **Routing decision process**: in chapter 5, the final probability distribution defining **the stationary state of a node** is used to make a decision on the nature of its behaviors. This process deserves more studies and specifications in order to increase the accuracy and **alleviate the uncertainty** of made decisions. We believe that this investigation can help to guarantee reliable packet routing decisions.

We also underline some long-term research directions that require more investigations in the future:

- **Complexity analysis**: the statistical and probabilistic techniques presented in chapter 4 and 5 should be further **studied and improved** in terms of complexity. We can also evaluate the performance of our solutions in a real MANET and then **optimize** the detection phases in order to minimize their **computational overhead**.

- **Resource constraints**: the **memory space** and **battery energy** required to store and process the collected nodes' behaviors and evolution information deserve more analysis in terms of resource consumption. This latter can be evaluated in a real testbed in order to design an adequate resource-aware detection solution of packet dropping attacks in MANETs.

- **Real implementation**: the solutions that we proposed in this thesis are able to detect different types of packet dropping attacks. However, we need to investigate their adaptability to be implemented as a **decentralized detection system** at each network node. We can also adapt our solutions as a **security extension** for an existing ad hoc routing protocol.

- **Cooperative packet dropping**: in this thesis we handled the case of malicious nodes that perform different types of dropping attacks without any cooperation between them. Based on recent studies, dropping attacks performed by **multiple nodes** in a cooperative manner can disrupt the routing services and still in the network for a long time period without being detected. We aim to investigate these attacks by studying their techniques and defining a model for their behaviors.

Bibliography

[1] J. Wan, D. Li, C. Zou, and K. Zhou. M2M communications for smart city: An event-based architecture. In *IEEE* 12^{th} *International Conference on Computer and Information Technology (CIT)*, pages 895–900. IEEE, 2012. (Cited on pages i and 11.)

[2] M. Frodigh, P. Johansson, and P. Larsson. Wireless ad hoc networking: the art of networking without a network. *Ericsson Review*, 4(4):249, 2000. (Cited on pages i and 11.)

[3] L. Abusalah, A. Khokhar, and M. Guizani. A survey of secure mobile ad hoc routing protocols. *IEEE communications surveys & tutorials*, 10(4):78–93, 2008. (Cited on pages ii and 12.)

[4] C. Li, Z. Wang, and C. Yang. SEAODV: A Security Enhanced AODV routing protocol for wireless mesh networks. In *Transactions on computational science XI*, pages 1–16. Springer, 2010. (Cited on pages ii, xi, 41 and 52.)

[5] Y. Ren, M. C. Chuah, J. Yang, and Y. Chen. Detecting blackhole attacks in Disruption-Tolerant Networks through packet exchange recording. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, pages 1–6, June 2010. (Cited on pages ii, 12, 41 and 52.)

[6] J.-. Cho, A. Swami, and R. Chen. A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 13(4):562–583, 2011. (Cited on pages iii and 12.)

[7] B. Wang, X. Chen, and W. Chang. A light-weight trust-based QoS routing algorithm for ad hoc networks. *Pervasive and Mobile Computing*, 13:164–180, August 2014. (Cited on pages iii, xi, 13, 45, 52 and 53.)

[8] S. Djahel, F. Nait-Abdesselam, and Z. Zhang. Mitigating packet dropping problem in mobile ad hoc networks: proposals and challenges. *IEEE communications surveys & tutorials*, 13(4):658–672, 2011. (Cited on pages iii and 13.)

[9] S. Şen and J. A. Clark. Intrusion detection in mobile ad hoc networks. In *Guide to wireless ad hoc networks*, pages 427–454. Springer, 2009. (Cited on pages iv and 14.)

[10] S. Zeadally, R. Hunt, Y. Chen, A. Irwin, and A. Hassan. Vehicular Ad Hoc Networks (VANETs): Status, Results, and Challenges. *Telecommunication Systems*, 50(4):217–241, 2012. (Cited on pages v and 19.)

[11] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell. Urban sensing systems: opportunistic or participatory ?. In *Proceedings of the* 9^{th} *workshop on Mobile computing systems and applications*, pages 11–16. ACM, 2008. (Cited on pages vi and 20.)

[12] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032 – 3080, 2011. (Cited on pages vi, 21 and 24.)

[13] R. C. Braley, I.C. Gifford, and R. F. Heile. Wireless Personal Area Networks: An Overview of the IEEE P802.15 Working Group. *Mobile Computing and Communications Review*, 4(1):26–33, January 2000. (Cited on pages vi and 22.)

[14] K. Chandrasekaran. *Essentials of Cloud Computing*. CRC Press, December 2014. (Cited on pages vi and 21.)

[15] P. Bonato. Advances in wearable technology and applications in physical medicine and rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 2(1):2, 2005. (Cited on pages vi and 22.)

[16] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing (AODV). In *Proceedings of the* 2^{nd} *IEEE Workshop on Mobile Computing Systems and Applications, WMCSA '99*, pages 90–100, February 1999. (Cited on pages vii, xii, 24, 53 and 54.)

[17] M. G. Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. In *Proceedings of the* 1^{st} *ACM Workshop on Wireless Security*, WiSE '02, pages 1–10, New York, NY, USA, 2002. ACM. (Cited on pages xi, 12, 39 and 52.)

[18] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E.M. Belding-Royer. A secure routing protocol for ad hoc networks. In 10^{th} *IEEE International Conference on Network Protocols, 2002. Proceedings*, pages 78–87, November 2002. (Cited on pages xi, 12, 38 and 52.)

[19] Rajiv K. Nekkanti and C.-W. Lee. Trust Based Adaptive on Demand Ad Hoc Routing Protocol. In *Proceedings of the* 42^{nd} *Annual Southeast Regional Conference*, ACM-SE 42, pages 88–93, New York, NY, USA, 2004. ACM. (Cited on pages xi, 40 and 52.)

[20] J. Yin and S. K. Madria. A hierarchical secure routing protocol against black hole attacks in sensor networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006*, volume 1, June 2006. (Cited on pages xi, 12, 39 and 52.)

[21] N. Chatterjee and J. K. Mandal. Detection of Blackhole Behaviour Using Triangular Encryption in NS2. *Procedia Technology*, 10:524–529, 2013. (Cited on pages xi, 12, 40 and 52.)

[22] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *Proceedings of the* 6^{th} *Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 255–265, New York, NY, USA, 2000. ACM. (Cited on pages xi, 43, 52 and 70.)

[23] P. Michiardi and R. Molva. Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. In *Proceedings of the IFIP TC6/TC11* 6^{th} *Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, pages 107–121, Deventer, The Netherlands, 2002. Kluwer, B.V. (Cited on pages xi, 12, 43 and 52.)

[24] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol. In *Proceedings of the* 3^{rd} *ACM International Symposium on Mobile Ad Hoc Networking & Computing*, MobiHoc '02, pages 226–236, New York, NY, USA, 2002. ACM. (Cited on pages xi, 12, 43, 44, 52 and 53.)

[25] X. Li, M. R. Lyu, and J. Liu. A trust model based routing protocol for secure ad hoc networks. In *2004 IEEE Aerospace Conference, 2004. Proceedings*, volume 2, pages 1286–1295, March 2004. (Cited on pages xi, 46, 52 and 53.)

[26] T. Eissa, S. A. Razak, R. H. Khokhar, and N. Samian. Trust-Based Routing Mechanism in MANET: Design and Implementation. *Mobile Networks and Applications*, 18:666–677, June 2011. (Cited on pages xi, 13, 47 and 52.)

[27] B. Wang, S. Soltani, J.K. Shapiro, and P.-N. Tan. Local detection of selfish routing behavior in ad hoc networks. In *Proceedings of the* 8^{th} *International Symposium on Parallel Architectures, Algorithms and Networks, ISPAN '2005.*, pages 392–399, December 2005. (Cited on pages xii, 48 and 52.)

[28] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and A. Nemoto. Detecting blackhole attack on aodv-based mobile ad hoc networks by dynamic learning method. *International Journal of Network Security*, 5(3):338–346, November 2007. (Cited on pages xii, 49, 52 and 53.)

[29] H. Xia, Z. Jia, X. Li, L. Ju, and E. H.-M. Sha. Trust prediction and trust-based source routing in mobile ad hoc networks. *Ad Hoc Networks*, 11(7):2096–2114, September 2013. (Cited on pages xii, 50 and 52.)

[30] L. Sanchez-Casado, G. Macia-Fernandez, P. Garcia-Teodoro, and R. Magan-Carrion. A model of data forwarding in MANETs for lightweight detection of malicious packet dropping. *Computer Networks*, 87:44–58, July 2015. (Cited on pages xii, 51, 52, 53 and 92.)

[31] V. Balakrishnan, V. Varadharajan, U. Tupakula, and M. E. G. Moe. Mitigating Flooding Attacks in Mobile Ad-hoc Networks Supporting Anonymous Communications. In *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications, 2007. AusWireless 2007*, August 2007. (Cited on pages xii, 54 and 59.)

[32] J.-H. Song, F. Hong, and Y. Zhang. Effective filtering scheme against RREQ flooding attack in mobile ad hoc networks. In *Proceedings of the 7th International Conference on Parallel and Distributed Computing, Applications and Technologies*, PDCAT '06, pages 497–502, Washington, DC, USA, 2006. IEEE Computer Society. (Cited on pages xii and 55.)

[33] S. Desilva and R. V. Boppana. Mitigating malicious control packet floods in ad hoc networks. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 4, pages 2112–2117, March 2005. (Cited on pages xii, 55 and 59.)

[34] P. Yi, Z. Dai, Y.-P. Zhong, and S. Zhang. Resisting flooding attacks in ad hoc networks. In *International Conference on Information Technology: Coding and Computing, 2005. ITCC 2005*, volume 2, pages 657–662, April 2005. (Cited on pages xii, 55 and 59.)

[35] F.-C. Jiang, C.-H. Lin, and H.-W. Wu. Lifetime elongation of ad hoc networks under flooding attack using power-saving technique. *Ad Hoc Networks*, 21:84–96, October 2014. (Cited on pages xii and 56.)

[36] R. Venkataraman, M. Pushpalatha, and T. Rama Rao. Performance Analysis of Flooding Attack Prevention Algorithm in MANETs. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 3(8):2056–2059, 2009. (Cited on pages xii, 56 and 59.)

[37] M. Abdelshafy and P. J. B. King. Resisting flooding attacks on AODV. In 8th *International Conference on Emerging Security Information, Systems and Technologies (SECUREWARE)*, pages 14–19, 2014. (Cited on pages xii and 57.)

[38] C. Panos, C. Xenakis, P. Kotzias, and I. Stavrakakis. A specification-based intrusion detection engine for infrastructure-less networks. *Computer Communications*, 54:67–83, December 2014. (Cited on pages xii, 57 and 59.)

[39] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: a defense against Wormhole attacks in wireless networks. In *INFOCOM 2003. 22nd Annual*

Joint Conference of the IEEE Computer and Communications. IEEE Societies, volume 3, pages 1976–1986, March 2003. (Cited on pages xiii, 59, 61 and 65.)

[40] J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos. TrueLink: A Practical Countermeasure to the Wormhole Attack in Wireless Networks. In *Proceedings of the 2006* 14^{th} *IEEE International Conference on Network Protocols, 2006. ICNP '06*, pages 75–84, November 2006. (Cited on pages xiii and 60.)

[41] H. Sun Chiu and K.-S. Lui. DelPHI: Wormhole detection mechanism for ad hoc wireless networks. In *2006* 1^{st} *International Symposium on Wireless Pervasive Computing*, January 2006. (Cited on pages xiii and 60.)

[42] V. K. Raju and K. V. Kumar. A simple andefficient mechanism to detect and avoid wormhole attacks in mobile ad hoc networks. In *2012 International Conference on Computing Sciences (ICCS)*, pages 271–275, September 2012. (Cited on pages xiii, 60 and 65.)

[43] S. Capkun, L. Buttyan, and J.-P. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *Proceedings of the* 1^{st} *ACM Workshop on Security of Ad Hoc and Sensor Networks*, SASN '03, pages 21–32, New York, NY, USA, 2003. ACM. (Cited on pages xiii and 61.)

[44] L. Hu and D. Evans. Using directional antennas to prevent Wormhole attacks. In *Proceedings of the* 11^{th} *Network and Distributed System Security Symposium*, pages 131–141, 2004. (Cited on pages xiii and 62.)

[45] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. W. Chang. Preventing Wormhole Attacks on Wireless Ad Hoc Networks: A Graph Theoretic Approach. In *2005 IEEE Wireless Communications and Networking Conference*, volume 2, pages 1193–1199, March 2005. (Cited on pages xiii, 33 and 62.)

[46] I. Khalil, S. Bagchi, and N. B. Shroff. LiteWorp: Detection and isolation of the Wormhole attack in static multihop wireless networks. *Computer Networks*, 51(13):3750–3772, September 2007. (Cited on pages xiii and 63.)

[47] I. Khalil, M. Awad, and A. Khreishah. CTAC: Control traffic tunneling attacks' countermeasures in mobile wireless networks. *Computer Networks*, 56(14):3300–3317, June 2012. (Cited on pages xiii and 63.)

[48] K. Govindan and P. Mohapatra. Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey. *IEEE Communications Surveys Tutorials*, 14(2):279–298, 2012. (Cited on pages xxiii, 92 and 117.)

[49] M. Rmayti, Y. Begriche, R. Khatoun, L. Khoukhi, X. Chen, and D. Gaiti. Wormhole attack detection using graph theory. In 10^{th} *edition of French*

International Seminar on Networks and Services Managment (GRES), Paris, France, December 2014. (Cited on pages xxiv and 118.)

[50] M. Rmayti, Y. Begriche, R. Khatoun, L. Khoukhi, and D. Gaiti. Flooding attacks detection in manets. In *International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*, pages 1–6, August 2015. (Cited on pages xxiv and 118.)

[51] A. K. Jain and V. Tokekar. Classification of denial of service attacks in mobile ad hoc networks. In *2011 International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 256–261. IEEE, 2011. (Cited on page 12.)

[52] S. Tan, X. Li, and Q. Dong. Trust based routing mechanism for securing oslr-based MANET. *Ad Hoc Networks*, 30:84 – 98, 2015. (Cited on pages 13, 48 and 52.)

[53] R. Lu, X. Lin, H. Zhu, and X. Shen. SPARK: A New VANET-based Smart Parking Scheme for Large Parking Lots. In *INFOCOM 2009, IEEE*, pages 1413–1421, April 2009. (Cited on page 20.)

[54] M. Buddhikot, G. Chandranmenon, S. Han, Y. W. Lee, S. Miller, and L. Salgarelli. Integration of 802.11 and third-generation wireless data networks. In *Proceedings of the* 22^{nd} *Annual Joint Conference of the IEEE Computer and Communications. INFOCOM 2003. IEEE Societies*, volume 1, pages 503–512 vol.1, March 2003. (Cited on page 20.)

[55] K. V. Sairam, N. Gunasekaran, and S. Redd. Bluetooth in wireless communication. *Communications Magazine, IEEE*, 40(6):90–96, 2002. (Cited on page 20.)

[56] Y. Yang, H. Hu, J. Xu, and G. Mao. Relay technologies for WiMax and LTE-advanced mobile systems. *IEEE Communications Magazine*, 47(10):100–105, October 2009. (Cited on page 20.)

[57] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas. LTE-advanced: next-generation wireless broadband technology [invited paper]. *IEEE Wireless Communications*, 17(3):10–22, June 2010. (Cited on page 20.)

[58] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver. Google street view: Capturing the world at street level. *Computer*, (6):32–38, 2010. (Cited on page 21.)

[59] M. Diao, Y. Zhu, J. Ferreira, and C. Ratti. Inferring individual daily activities from mobile phone traces: A boston example. *Environment and Planning B: Planning and Design*, 2015. (Cited on page 21.)

[60] E. Paulos and T. Jenkins. Urban probes: encountering our emerging urban atmospheres. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 341–350. ACM, 2005. (Cited on page 21.)

[61] N. Skeledzija, J. Cesic, E. Koco, V. Bachler, H. N. Vucemilo, and H. Džapo. Smart home automation system for energy efficient housing. In *2014* 37^{th} *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 166–171, May 2014. (Cited on page 21.)

[62] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005. (Cited on page 21.)

[63] M. L. Sichitiu. Wireless mesh networks: opportunities and challenges. In *Proceedings of World Wireless Congress*, volume 2, 2005. (Cited on page 21.)

[64] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *International workshop on wearable and implantable body sensor networks*, volume 5, 2004. (Cited on page 22.)

[65] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano. Device-to-device communications with Wi-Fi Direct: overview and experimentation. *Wireless Communications, IEEE*, 20(3):96–104, 2013. (Cited on page 22.)

[66] V. Coskun, B. Ozdenizci, and K. Ok. A survey on Near Field Communication (NFC) technology. *Wireless Personal Communications*, 71(3):2259–2294, August 2013. (Cited on page 23.)

[67] J. Pan, S. Paul, and R. Jain. A survey of the research on future internet architectures. *Communications Magazine, IEEE*, 49(7):26–36, 2011. (Cited on page 23.)

[68] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *SIGCOMM Computer Communication Review*, 24(4):234–244, October 1994. (Cited on page 23.)

[69] C. R. Lin. QoS routing in ad hoc wireless networks. In *Proceedings of the* 23^{rd} *Annual Conference on Local Computer Networks, LCN '98*, pages 31–40, October 1998. (Cited on page 23.)

[70] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi. A framework for reliable routing in mobile ad hoc networks. In *INFOCOM 2003.* 22^{nd} *Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 270–280, March 2003. (Cited on page 23.)

[71] S. Yi, P. Naldurg, and R. Kravets. Security-aware ad hoc routing for wireless networks. In *Proceedings of the* 2^{nd} *ACM International Symposium on Mobile*

Ad Hoc Networking & Computing, MobiHoc '01, pages 299–302, New York, NY, USA, 2001. ACM. (Cited on page 23.)

[72] J. Loo, J. Lloret Mauri, and J. H. Ortiz. *Mobile Ad hoc networks: current status and future trends*. CRC Press, 2011. (Cited on page 24.)

[73] Z. J. Haas, M. R. Pearlman, and P. Samar. The zone routing protocol (zrp) for ad hoc networks. 2002. (Cited on page 24.)

[74] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer. Sharp: A hybrid adaptive routing protocol for mobile ad hoc networks. In *Proceedings of the* 4^{th} *ACM international symposium on Mobile ad hoc networking & computing*, pages 303–314. ACM, 2003. (Cited on page 24.)

[75] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR), 2003. (Cited on pages 24 and 25.)

[76] A. Huhtonen. Comparing AODV and OLSR routing protocols. *Telecommunications Software and Multimedia*, pages 1–9, 2004. (Cited on page 25.)

[77] A. N. Thakare and M. Joshi. Performance analysis of AODV & DSR routing protocol in mobile ad hoc networks. *IJCA Special Issue on MANETs*, 4:211–218, 2010. (Cited on page 25.)

[78] A. K. Gupta, H. Sadawarti, and A. K. Verma. Performance analysis of AODV, DSR & TORA routing protocols. *International Journal of Engineering and Technology*, 2(2):226, 2010. (Cited on page 25.)

[79] H. Zhang, M. Bialkowski, G. Einicke, and J. Homer. An extended AODV protocol for VoIP application in mobile ad hoc network. In *International Symposium on Communications and Information Technologies (ISCIT'07)*, pages 836–841. IEEE, 2007. (Cited on page 25.)

[80] V. Silva, L. A. Guedes, and F. Vasques. A new AODV-based routing protocol adequate for monitoring applications in oil & gas production environments. In *2010* 8^{th} *IEEE international workshop on Factory communication systems (WFCS)*, pages 283–292. IEEE, 2010. (Cited on page 25.)

[81] R. W. Shirey. Internet Security Glossary, Version 2. Network Working Group, August 2007. (Cited on page 26.)

[82] F. Xing and W. Wang. Understanding Dynamic Denial of Service Attacks in Mobile Ad Hoc Networks. In *IEEE Military Communications Conference, 2006. MILCOM 2006*, pages 1–7, October 2006. (Cited on page 26.)

[83] A. D. Wood and J. A. Stankovic. Handbook of sensor networks: Compact wireless and wired sensing systems, 2005. (Cited on page 26.)

[84] T.-S. Chou. Security threats on cloud computing vulnerabilities. *International Journal of Computer Science and Information Technology*, 5(3):79–88, June 2013. (Cited on page 27.)

[85] G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer, and R. A. Kemmerer. An Intrusion Detection Tool for AODV-based Ad hoc Wireless Networks. In *Proceedings of the* 20^{th} *Annual Computer Security Applications Conference*, ACSAC '04, pages 16–27, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on page 27.)

[86] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala. DoS Attacks in Mobile Ad Hoc Networks: A Survey. In *Proceedings of the* 2^{nd} *International Conference on Advanced Computing Communication Technologies (ACCT)*, pages 535–541, January 2012. (Cited on page 28.)

[87] A. Alsumayt and J. Haggerty. A survey of the mitigation methods against DoS attacks on MANETs. In *Science and Information Conference (SAI), 2014*, pages 538–544, August 2014. (Cited on page 28.)

[88] B. Schneier. *Secrets & Lies: Digital Security in a Networked World*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2000. (Cited on page 28.)

[89] F.-H. Tseng, L.-D. Chou, and H.-C. Chao. A survey of black hole attacks in wireless mobile ad hoc networks. *Human-centric Computing and Information Sciences*, 1(1):1–16, 2011. (Cited on page 29.)

[90] I. Aad, J.-P. Hubaux, and E. W. Knightly. Denial of Service Resilience in Ad Hoc Networks. In *Proceedings of the* 10^{th} *Annual International Conference on Mobile Computing and Networking*, MobiCom '04, pages 202–215, New York, NY, USA, 2004. ACM. (Cited on page 29.)

[91] P. Michiardi. *Mécanismes de sécurité et de coopération entre nœuds d'un réseaux mobile ad hoc*. PhD thesis, Télécom ParisTech, 2004. (Cited on page 29.)

[92] H. Kim, R. B. Chitti, and J. Song. Novel defense mechanism against data flooding attacks in wireless ad hoc networks. *IEEE Transactions on Consumer Electronics*, 56(2):579–582, 2010. (Cited on page 32.)

[93] M. Pirretti, S. Zhu, N. Vijaykrishnan, P. McDaniel, M. Kandemir, and R. Brooks. The sleep deprivation attack in sensor networks: Analysis and methods of defense. *International Journal of Distributed Sensor Networks*, 2(3):267–287, 2006. (Cited on page 32.)

[94] Y.-C. Hu, A. Perrig, and D. B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the* 2^{nd} *ACM Workshop on Wireless Security*, number 11 in WiSe '03, pages 30–40, New York, NY, USA, 2003. ACM. (Cited on page 33.)

[95] M. Khabbazian, H. Mercier, and V.K. Bhargava. Severity analysis and countermeasure for the Wormhole attack in wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 8(2):736–745, February 2009. (Cited on page 33.)

[96] M. Meghdadi, S. Ozdemir, and I. Güler. A survey of Wormhole-based attacks and their countermeasures in wireless sensor networks. *IETE technical review*, 28(2):89–102, 2011. (Cited on page 33.)

[97] K. Saghar, D. Kendall, and A. Bouridane. RAEED: A solution for hello flood attack. In *2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 248–253, January 2015. (Cited on page 34.)

[98] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili. A pairwise key predistribution scheme for wireless sensor networks. *ACM Trans. Inf. Syst. Secur.*, 8(2):228–258, May 2005. (Cited on page 41.)

[99] B. Solha, D. Elgesem, and K. Stolen. Why Trust is not Proportional to Risk. In *The 2nd International Conference on Availability, Reliability and Security. ARES 2007*, pages 11–18, April 2007. (Cited on page 42.)

[100] I. M Atakli, H. Hu, Y. Chen, W. S. Ku, and Z. Su. Malicious node detection in wireless sensor networks using weighted trust evaluation. In *Proceedings of the 2008 Spring simulation multiconference*, pages 836–843. Society for Computer Simulation International, 2008. (Cited on page 42.)

[101] C. Zhang, X. Zhu, Y. Song, and Y. Fang. A formal study of trust-based routing in wireless ad hoc networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010. (Cited on page 42.)

[102] K. Mandalas, D. Flitzanis, G.F. Marias, and P. Georgiadis. A survey of several cooperation enforcement schemes for manets. In *Proceedings of the 5th IEEE International Symposium on Signal Processing and Information Technology, 2005.*, pages 466–471. IEEE, 2005. (Cited on page 42.)

[103] S. Buchegger and J.-Y. Le Boudec. A robust reputation system for mobile ad-hoc networks. Technical report, 2003. (Cited on page 42.)

[104] D. B. Johnson, D. A. Maltz, and J. Broch. Ad hoc networking. chapter DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pages 139–172. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001. (Cited on pages 43, 53 and 54.)

[105] S. Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly Media, Inc., 1995. (Cited on page 44.)

[106] A. Mitrokotsa and C. Dimitrakakis. Intrusion detection in MANET using classification algorithms: The effects of cost and model selection. *Ad Hoc Networks*, 11(1):226–237, January 2013. (Cited on page 48.)

[107] B. Zhu, Z. Wan, M.S. Kankanhalli, F. Bao, and R.H. Deng. Anonymous secure routing in mobile ad-hoc networks. In 29^{th} *Annual IEEE International Conference on Local Computer Networks, 2004*, pages 102–108, November 2004. (Cited on page 54.)

[108] L. Buttyan, L. Dora, and I. Vajda. Statistical Wormhole Detection in Sensor Networks. In R. Molva, G. Tsudik, and D. Westhoff, editors, *Security and Privacy in Ad-hoc and Sensor Networks*, number 3813 in Lecture Notes in Computer Science, pages 128–141. Springer Berlin Heidelberg, 2005. (Cited on page 64.)

[109] N. Song, L. Qian, and X. Li. Wormhole Attacks Detection in Wireless Ad Hoc Networks: A Statistical Analysis Approach. In *Proceedings of the* 19^{th} *IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, volume 18 of *IPDPS '05*, Washington, DC, USA, April 2005. IEEE Computer Society. (Cited on page 64.)

[110] S. Song, H. Wu, and B.-Y. Choi. Statistical Wormhole detection for mobile sensor networks. In *2012* 4^{th} *International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 322–327, July 2012. (Cited on page 65.)

[111] M.-Y. Su. Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems. *Computer Communications*, 34(1):107 – 117, 2011. (Cited on pages 72 and 73.)

[112] J. J. Eberhardt. Bayesian spam detection. *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal*, 2(2):1–7, 2015. (Cited on pages 72 and 73.)

[113] A. Hertzmann. Introduction to Bayesian Learning. In *ACM SIGGRAPH 2004 Course Notes*, SIGGRAPH '04, New York, NY, USA, 2004. ACM. (Cited on page 76.)

[114] B. Ristic, B. T. Vo, B. N. Vo, and A. Farina. A Tutorial on Bernoulli Filters: Theory, Implementation and Applications. *IEEE Transactions on Signal Processing*, 61(13):3406–3430, July 2013. (Cited on page 76.)

[115] C. P. Robert. *Le choix Bayésien. Principes et pratiques*. Springer, 2006. (Cited on page 78.)

[116] P.D. Hoff. *A First Course in Bayesian Statistical Methods*. Springer, 2009. (Cited on pages 78 and 98.)

[117] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos, and P. Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. In *Workshop on Machine Learning and Textual Information Access*, pages 1–3, 2000. (Cited on page 85.)

[118] V. Laxmi, C. Lal, M. S. Gaur, and D. Mehta. JellyFish attack: Analysis, detection and countermeasure in TCP-based MANET. *Journal of Information Security and Applications*. (Cited on pages 92 and 95.)

[119] H. Xia, J. Yu, C.-L. Tian, Z.-K. Pan, and E. Sha. Light-weight trust-enhanced on-demand multi-path routing in mobile ad hoc networks. *Journal of Network and Computer Applications*, 62:112–127, February 2016. (Cited on page 92.)

[120] J. Luo, X. Liu, Y. Zhang, D. Ye, and Z. Xu. Fuzzy trust recommendation based on collaborative filtering for mobile ad-hoc networks. In *LCN*, pages 305–311. Citeseer, 2008. (Cited on page 99.)

[121] J. Zhao and B. K. Bose. Evaluation of membership functions for fuzzy logic controlled induction motor drive. In *IEEE 2002* 28^{th} *Annual Conference of the Industrial Electronics Society, IECON' 02.*, volume 1, pages 229–234. IEEE, 2002. (Cited on page 99.)

[122] Y. Ephraim and N. Merhav. Hidden markov processes. *IEEE Transactions on information theory*, 48(6):1518–1569, 2002. (Cited on page 102.)

[123] O. Sarig. Lecture notes on ergodic theory. *Penn State, University Park, Pennsylvania*, 12, 2008. (Cited on page 102.)

Publications

Journal articles

- Mohammad Rmayti, Youcef Begriche, Rida Khatoun and Dominique Gaiti. A Stochastic Approach for Packet Dropping Attacks Detection in Mobile Ad hoc Networks. *Journal of Computer Networks*. 2016. (*accepted*).

- Mohammad Rmayti, Rida Khatoun, Sherali Zeadally and Dominique Gaiti. Denial of Service in Ad hoc Networks: Security Attacks, Detection Mechanisms and Research Opportunities. (*under preparation*).

- Mohammad Rmayti, Youcef Begriche and Rida Khatoun. A Probabilistic Classification Framework for Packet Dropping Attacks Detection in MANETs. (*under preparation*).

International conferences

- Mohammad Rmayti, Youcef Begriche, Rida Khatoun, Lyes Khoukhi and Dominique Gaiti. Denial of service (DoS) Attacks Detection in MANETs using Bayesian Classifiers. In *Proceedings of the* 21^{st} *IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT'14)*, Delft, The Netherlands. November 2014. (*Best Student Paper Award*).

- Mohammad Rmayti, Youcef Begriche, Rida Khatoun, Lyes Khoukhi and Dominique Gaiti. Denial of Service (DoS) Attacks Detection in MANETs through Statistical Models. In *Proceedings of Global Information Infrastructure and Networking Symposium (GIIS'14)*, Montréal, Canada. September 2014.

- Mohammad Rmayti, Youcef Begriche, Rida Khatoun, Lyes Khoukhi and Dominique Gaiti. Flooding attacks detection in MANETs. In *Proceedings of the 2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC'2015)*, Shanghai, China. August 2015.

National conferences

- Mohammad Rmayti, Youcef Begriche, Rida Khatoun, Lyes Khoukhi, Dominique Gaiti and Chen Xiuzhen. Wormhole Attacks Detection in MANETs using Graph Theory. In *Proceedings of the* 10^{th} *edition of the French Seminar for Networks Management and Services (GRES'14).* Paris, France. December 2014.

Mohammad RMAYTI

Doctorat : Ingénierie Sociotechnique des Connaissances, des Réseaux et du Développement Durable

Année 2016

Une approche décentralisée pour la détection de comportements malveillants dans les réseaux MANETs

Avec l'évolution des besoins d'utilisateurs, plusieurs technologies de réseaux sans fil ont été développées. Parmi ces technologies, nous trouvons les réseaux mobiles ad hoc (MANETs) qui ont été conçus pour assurer la communication dans le cas où le déploiement d'une infrastructure réseaux est coûteux ou inapproprié. Dans ces réseaux, le routage est une fonction primordiale où chaque entité mobile joue le rôle d'un routeur et participe activement dans le routage. Cependant, les protocoles de routage ad hoc tel qu'ils sont conçus manquent de contrôle de sécurité. Sur un chemin emprunté, un nœud malveillant peut violemment perturber le routage en bloquant le trafic. Dans cette thèse, nous proposons une solution de détection des nœuds malveillants dans un réseau MANET basée sur une analyse comportementale à travers les filtres bayésiens et les chaînes de Markov. L'idée de notre solution est d'évaluer le comportement d'un nœud en fonction de ses échanges avec ses voisins d'une manière complètement décentralisée. Par ailleurs, un modèle stochastique est utilisé afin de prédire la nature de comportement d'un nœud et vérifier sa fiabilité avant d'emprunter un chemin. Notre solution a été validée via de nombreuses simulations sur le simulateur NS-2. Les résultats montrent que la solution proposée permet de détecter avec précision les nœuds malveillants et d'améliorer la qualité de services de réseaux MANETs.

Mots clés : réseaux ad hoc (informatique) - attaques par déni de service - réseaux d'ordinateurs, mesures de sûreté - statistique bayésienne – processus stochastiques.

Misbehaviors Detection Schemes in Mobile Ad Hoc Networks

With the evolution of user requirements, many network technologies have been developed. Among these technologies, we find mobile ad hoc networks (MANETs) that were designed to ensure communication in situations where the deployment of a network infrastructure is expensive or inappropriate. In this type of networks, routing is an important function where each mobile entity acts as a router and actively participates in routing services. However, routing protocols are not designed with security in mind and often are very vulnerable to node misbehavior. A malicious node included in a route between communicating nodes may severely disrupt the routing services and block the network traffic. In this thesis, we propose a solution for detecting malicious nodes in MANETs through a behavior-based analysis and using Bayesian filters and Markov chains. The core idea of our solution is to evaluate the behavior of a node based on its interaction with its neighbors using a completely decentralized scheme. Moreover, a stochastic model is used to predict the nature of behavior of a node and verify its reliability prior to selecting a path. Our solution has been validated through extensive simulations using the NS-2 simulator. The results show that the proposed solution ensures an accurate detection of malicious nodes and improve the quality of routing services in MANETs.

Keywords: ad hoc networks (computer networks) - denial of service attacks - computer networks, security measures - Bayesian statistical decision theory – stochastic processes.