UNIVERSITÉ DE MONTRÉAL

MODÉLISATION DES ATTAQUANTS ET DE LA RÉPUTATION DANS LES RÉSEAUX AD HOC MOBILES

SÉBASTIEN MAXIME RIVARD
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE
MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
NOVEMBRE 2007

Your file Votre référence
ISBN: 978-0-494-36936-4
Our file Notre référence
ISBN: 978-0-494-36936-4

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MODÉLISATION DES ATTAQUANTS ET DE LA RÉPUTATION DANS LES RÉSEAUX AD HOC MOBILES

présenté par : RIVARD Sébastien Maxime
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury constitué de :

M. LANGLOIS Pierre, Ph.D., président
M. PIERRE Samuel, Ph.D., membre et directeur de recherche
M. FERNANDEZ José Manuel, Ph.D., membre et codirecteur de recherche
M. QUINTERO Alejandro, Doct., membre

À Geneviève, pour ton appui infaillible.

Remerciements

Tout d'abord, j'aimerais remercier mon directeur de recherche, M. Samuel Pierre, professeur titulaire au département de génie informatique de l'École Polytechnique de Montréal pour son aide tant financière qu'académique. Je tient également à remercier mon codirecteur, M. José Fernandez, professeur adjoint au département de génie informatique de l'École Polytechnique de Montréal. C'est grâce à leurs précieux conseils que j'ai pu mener à bien ce mémoire de maîtrise.

Aussi, je remercie messieurs Stéphane Ouellette et Angelo Rossi qui m'ont aidé dans mes travaux.

Je tiens également à remercier messieurs Pierre-Marc Bureau, Georges Abou-Khalil, Claude-Olivier Pitt et Kamel Ayari qui ont été d'excellents partenaires pour les laboratoires et projets.

Je remercie aussi les membres et le personnel du LARIM, entre autres Gabriel Ioan Ivascu et Marc Doyon.

Bien sûr, je remercie mes parents et amis pour leur support indéfectible.

Résumé

Les réseaux ad hoc mobiles (RAHM) sont des réseaux où les nœuds sont usagers, serveurs et routeurs. Cette structure décentralisé entraîne de nombreux problèmes de sécurités, tels la non-collaboration des nœuds. Ceux-ci peuvent laisser tomber les paquets qu'ils devraient acheminer. La réputation des nœuds est utilisée afin de pénaliser les attaquants et de les empêcher de dégrader la qualité de service offerte sur le réseau. Toutefois, l'analyse de performance des protocoles utilisant la réputation semble souvent inadéquate : l'attaquant, l'environnement et les critères de performances sont mal définis. Comme aucun modèle analytique ne permet de prévoir quel sera l'efficacité des contres-mesures utilisant la réputation (CMUR) dans un environnement précis, il est très difficile de prouver si elles sont réelement efficaces.

Ce mémoire a pour objectif principal de proposer un modèle décrivant la relation entre l'environnement des RAHM, les attaquants agissant dans les RAHM, ainsi que différents critères de performances tels le taux de faux positifs, le taux de faux négatifs et le taux de pertes de paquets. Ce modèle permettrait alors de connaître précisément la validité d'une CMUR.

Le modèle utilise les processus stochastiques afin de modéliser la connectivité entre les nœuds, le trafic, les flots entre les paires de nœuds, les différents types de pertes de paquets ainsi que différents indices de performance. Afin de pouvoir modéliser tous ces phénomènes, différentes hypothèses sont posées, les plus importantes étant que le trafic varie très lentement et que le temps inter-arrivée entre deux trames devant être transmises par un nœud suit une loi exponentielle. Comme certains des processus stochastiques sont décrits par des systèmes d'équations non-linéaires, il est impossible de résoudre analytiquement ce modèle. Des méthodes de simulations stochastiques sont donc utilisées afin de pouvoir le résoudre numériquement.

Le modèle a été partiellement implanté afin de simuler un RAHM utilisant le protocole de routage DSR et ne comprenant aucune CMUR. La validation du modèle est effectuée en comparant les résultats obtenus avec le modèle et ceux obtenus avec Qualnet, un logiciel commercial de simulation de réseaux. Il est démontré à l'aide du test statistique de Kolmogorov-Smirnov que le modèle ne peut pas reproduire exactement les résultats obtenus à l'aide de Qualnet. En effet, la probabilité qu'un

paquet soit jeté calculée par Qualnet est différente de celle calculée avec le modèle. Des résultats obtenus en simulant un cas particulier à l'aide de Qualnet montrent que le temps inter-arrivée entre les trames émises par les nœuds ne suit pas une distribution exponentielle ce qui peut expliquer cet écart. Ceci se produit même si la demande générée par l'application est exponentielle. Ce résultat est toutefois très important étant donné que l'hypothèse voulant que le temps inter-arrivé entre les trames soit exponentiel est très souvent utilisée dans la littérature.

Le modèle proposé est le premier qui décrit l'interaction entre les attaquants et leur milieu, les autres modèles ne prenant en compte que les attaquants ou l'environnement. Toutefois, il n'est pas suffisamment précis pour être utilisé afin de valider une CMUR. Lors de travaux subséquents, il sera donc nécessaire de modifier la portion du modèle décrivant les interférences dans les RAHM afin d'améliorer sa précision. De plus, seuls certains aspects du modèle ont été simulés. Il sera nécessaire de prendre en compte la mobilité ainsi que différentes CMUR. De plus, des leçons apprises sont données afin d'orienter d'éventuelles recherches dans la validation des CMUR.

Abstract

Mobile ad-hoc networks (MANET) are networks where nodes are users, servers and routers. This structure causes several security problems such as those caused by node greeniness. Nodes can drop packets which they should normally forward. Node reputation is used to identify and eliminate malicious nodes and prevent them from deteriorating network Quality of Service (QoS). However, performance analysis of reputation-based counter-measures (RBCM) is often flawed. The attackers, the environment and the performance parameters are not always well defined and there is no analytical model that can be used to compute the efficiency of RBCM.

The main objective of this Master's thesis is to define an analytical model that describes the interaction between MANET environment, attackers and performance criterias such as false positive rate, false negative rate and packet drop rate. With this model, it would be possible to know exactly the precision of a RBCM.

Stochastic processes are used to model node connectivity, traffic generation, transmissions between nodes, packets losses in order to compute quantitative metrics. Two main hypotheses are used. First, traffic changes very slowly. Second, the packet inter-arrival time distribution is exponential. Some of the stochastic processes are composed of nonlinear equations, which means that stochastic simulation methods must be used to resolve the model.

This model has been partly implemented to simulate a MANET using the DSR routing protocol without any RBCM. Performance analysis has been done by comparing the model results and those obtained with Qualnet, a commercial network simulator. The Kolmogorov-Smirnov statistical test demonstrates that the model cannot reproduce Qualnet results. There are big differences between the values for the probability to drop packets calculated with Qualnet and those calculated with the model. Special cases tested with Qualnet show that the inter-arrival time between frames that nodes must transmit is not exponentially distributed. This could explain the differences between the model results and those from Qualnet.

This is the first model that describes interactions between attackers and the environment. The other models only describe the environment or the attackers. However, it cannot be used to validate RBCM performance. In future work, it would be impor-

tant to improve collision modeling in order to increase model precision. Only some model aspects were validated in this Master's thesis. It will thus be important to validate all aspects in future work. We give several lessons learned to help other researchers validate RBCM.

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et abréviations

ANP	Assistant numérique personnel
CAM	Code d'authentication de message
CGA	Cryptographically Generated Adresses
CMUR	Contres-mesure utilisant la réputation
DSR	Dynamic Source Routing
IETF	Internet Engineering Task Force
MANET	Mobile Ad hoc Networks
PGP	Pretty Good Privacy
QoS	Quality of Service
RAHM	Réseau ad hoc mobile
RAP	Rushing Attack Prevention
RBCM	Reputation Based Counter-Measure
SDI	Système de détection d'intrus
TESLA	Timed Efficient Stream Loss-tolerant Authentication
VA	Variable aléatoire

Description des variables utilisées

J	VA déterminant si un attaquant jette un paquet
J'	VA déterminant si un attaquant ment au sujet de la réputation
Γ	Demande entre une source et une destination
PSR	Puissance du signal reçu par un nœud
PE	Puissance du siganl émis par un nœud mesurée à 1 m
FA	Facteur d'atténuation du signal radio
L	Distance entre deux nœuds
VAA	VA décrivant l'atténuation du signal radio
$STDA$	Valeur de l'écart type de l'atténuation du signal radio
SP	Seuil de détection du signal radio pour un nœud
C	Matrice décrivant la connectivité du réseau
Co	VA décrivant le trafic entre deux nœuds
γ	Flot moyen entre deux nœuds
W	Poid des arcs dans le graphe représentant le réseau
c	Fonction calculant le plus court chemin dans le réseau
I_1	Fonction caractérisant une CMUR
I_2	Fonction caractérisant une CMUR
I_3	Fonction caractérisant une CMUR
I_4	Fonction caractérisant une CMUR
I_5	Fonction caractérisant une CMUR
I_6	Fonction caractérisant une CMUR
I_7	Fonction caractérisant une CMUR
E	Fonction caractérisant une CMUR
T_{ACK}	Temps requis pour émettre un accusé réception 802.11
T_{DIFS}	Temps d'écoute requis pour le DIFS 802.11
T_{DATA}	Temps requis pour émettre une trame 802.11
S^{a}	VA décrivant une partie des collisions
S^{b}	VA décrivant une partie des collisions
S^{c}	VA décrivant une partie des collisions
S^{d}	VA décrivant une partie des collisions

D	VA décrivant si la détection s'est effectuée correctement
T	VA décrivant si un paquet a été jeté
TO	VA décrivant si un paquet a été observé comme étant jeté
F	VA décrivant si un paquet est jeté pour réputation trop faible
H	Route
NbH	Nombre de routes
A	VA décrivant si un accusé réception actif est reçu
v	Valeur prise par une réputation
V	Ensemble des valeurs que peuvent prendre les réputations
R^{tot}	Réputation globale
R^{obs}	Réputation basée sur les observations
R^{ack}	Réputation basée sur les accusés réception
R^{prop}	Réputation basée sur les avis des autres nœuds
$\bar{R}^{\text{tot}}$	Réputation globale, temps passé
$\bar{R}^{\text{obs}}$	Réputation basée sur les observations, temps passé
$\bar{R}^{\text{ack}}$	Réputation basée sur les accusés réception, temps passé
$\bar{R}^{\text{prop}}$	Réputation basée sur les avis des autres nœuds, temps passé
f^{c}	Facteur caractérisant l'entente entre les nœuds
λ^{DATA}	Partie du trafic causant les observations erronées
λ^{ACK}	Partie du trafic causant les observations erronées
$\bar{\lambda}^{\text{DATA}}$	Partie du trafic causant les observations erronées
$\bar{\lambda}^{\text{ACK}}$	Partie du trafic causant les observations erronées
nb	Variable décrivant le nombre de paquets
nb'	Constante décrivant le nombre de paquets
Λ^{a}	Partie du trafic causant les interférences
Λ^{b}	Partie du trafic causant les interférences
Λ^{c}	Partie du trafic causant les interférences
Λ^{d}	Partie du trafic causant les interférences
NH	VA définissant si un nœud est honnête
FP	Taux de faux positifs
FN	Taux de faux négatifs
$TPNM$	Taux de pertes de paquets, nœuds malicieux
$TPNH$	Taux de pertes de paquets, nœuds honnêtes

CHAPITRE 1

Introduction

Lors de déploiements militaires ou d'aide humanitaire, le contrôle des effectifs est un élément clef qui peut mener au succès ou à l'échec de la mission. C'est ce que les militaires appellent le commandement et contrôle. Dès leur apparition, les systèmes radio et informatiques sont devenus à leur tour un élément clef de toute planification, tant humanitaire que militaire. Ces technologies nécessitent souvent la mise en place d'une infrastructure de communication. Des relais sont dispersés à travers le théâtre d'opérations afin de permettre des communications à longue distance. Toutefois, il existe des situations ou ces relais ne peuvent être mis en place. Par exemple, lors de désastres naturels, l'infrastructure existante et servant aux services ambulanciers ou policiers pourrait être totalement détruite. Remettre en place cette infrastructure prend du temps et divertit des ressources qui pourraient autrement venir en aide à la population. Il est donc souhaitable d'avoir des moyens de communications qui ne nécessitent pas d'infrastructure préalable.

Les réseaux ad hoc mobiles sont apparus dernièrement. Toutes les fonctions normalement assumées par une infrastructure sont assurées par les utilisateurs mobiles. Ces réseaux sont très utiles en cas de désastres, pour les réseaux militaires ou dans les points de services. Ils peuvent être déployés sans coût d'infrastructure. Ils sont toutefois très vulnérables à une panoplie d'attaques. Malgré des protocoles de plus en plus sécuritaires, il est encore possible de corrompre ces réseaux. De nombreux protocoles de routage sécuritaires ont été proposés et des analyses de performance de ceux-ci ont été effectuées. Par contre, la validité de ces analyses est difficile à vérifier étant donné l'absence de modèle analytique pertinent. Ce mémoire vise donc à apporter un modèle représentatif des interactions entre les attaquants et l'environnement du réseau ad hoc mobile.

Dans ce chapitre d'introduction, nous présentons les différents concepts de base des réseaux ad hoc mobiles (RAHM). Ensuite, nous présentons les différents éléments de la problématique, suivis par les objectifs de recherche. Finalement, le plan du

mémoire est présenté.

1.1 Définitions et concepts de base

Un réseau ad hoc mobile (RAHM) est un réseau sans fil qui n'utilise aucune infrastructure fixe. Tous les paquets sont acheminés par les usagers qui sont donc à la fois utilisateurs de services et routeurs. Ainsi, il leur est possible de communiquer entre eux à des distances supérieures à leur portée d'émission.

Cette façon de faire est très utile lors de catastrophes naturelles. Les différents intervenants sont capables de communiquer entre eux sans avoir à réparer des installations détruites et nécessaires aux réseaux sans fils conventionnels. De plus, il est possible de bâtir des points de services sans avoir à implanter d'infrastructure ni d'entité administrative.

Différents protocoles de routage ont été développés. Ceux-ci sont classifiés en deux familles de protocoles : les protocoles proactifs et les protocoles réactifs. La première catégorie comprend les protocoles qui créent les tables de routage avant que les nœuds ne tentent d'envoyer des paquets sur le réseau tandis que la seconde ne crée ces tables que lorsqu'il y a un paquet à être acheminé vers une destination dont la route est inconnue.

Malheureusement, ces réseaux sont beaucoup plus exposés à des attaques de dénis de service. Tant les attaquants potentiels que les usagers légitimes utilisent le même canal. De plus, aucune autorité centrale n'est présente. Sans serveurs ni routeurs, les mécanismes de protection classiques, tels les pare-feux, ne peuvent être implémentés.

Il est donc nécessaire de développer des protocoles de routage capables de résister aux divers attaquants. Ils visent à assurer l'intégrité de l'information contenue dans les tables de routage. Mais ces protocoles sont sujets à diverses contraintes. La capacité des batteries des nœuds, leur puissance de calcul et la bande passante disponible sont très limitées. Il est donc nécessaire que ces mécanismes respectent ces contraintes. L'utilisation de la cryptographie asymétrique est donc moins envisageable, ce qui entraîne d'autres défis. L'absence d'autorité centrale rend plus difficile la distribution des certificats et impose d'autres restrictions aux protocoles.

Plusieurs protocoles de routage sécuritaire ont été développés au cours des dernières années. Ceux-ci, la plupart du temps basés sur des protocoles qui ne sont pas sécuritaires, authentifient les différents paquets à l'aide de signatures. Celles-ci sont

basées sur les clefs asymétriques ou sur des protocoles comme TESLA (Perrig *et al.*, 2002). Les protocoles de routage sécuritaire arrivent à assurer une convergence des tables de routage et à assurer leur conformité au protocole.

Bien que les protocoles de routage sécuritaire peuvent assurer l'intégrité des tables de routage, les nœuds ne sont pas à priori forcés d'acheminer le trafic conformément à celles-ci. Selon certains auteurs (Yang *et al.*, 2004), aucun mécanisme visant à prévenir les attaques ne pourra les en empêcher. Il est donc nécessaire de détecter les nœuds qui ne collaborent pas, communément appelés « nœuds égoïstes ». La théorie des jeux vise à modéliser les interactions entre des acteurs agissant dans leur propre intérêt. Elle peut donc être utilisée afin de modéliser ces nœuds. Des mécanismes de contrôle s'inspirant de la théorie des jeux, tels une monnaie (Buttyàn et Hubaux, 2001) ou un système utilisant la réputation (Marti *et al.*, 2000; Buchenner et Le Boudec, 2002a; Hasswa *et al.*, 2005; Michiardi et Molva, 2002), ont été proposés. Ces derniers visent à repérer les nœuds malicieux ou égoïstes et à les exclure du réseau. Ainsi, ils ne peuvent plus faire acheminer leurs paquets ni faire tomber ceux des autres. Mais ces systèmes ne sont pas à l'abri des attaquants, leurs faiblesses pouvant être utilisées afin d'effectuer des dénis de services. Il serait possible pour un nœud malicieux de faire croire aux nœuds honnêtes qu'un des leurs est égoïste. Ainsi, il serait exclu du réseau injustement. Il est aussi possible pour les nœuds malicieux de passer inaperçus.

1.2 Éléments de la problématique

L'égoïsme est à lui seul une menace sérieuse envers le bon fonctionnement des RAHM. Ce comportement est le reflet d'une caractéristique intrinsèque de ces réseaux : les unités mobiles tentent de diminuer au maximum leur consommation d'énergie.La durée de vie des batteries des nœuds étant grandement limitée, acheminer les paquets des autres est une dépense énergétique dont bien des unités mobiles se passeraient.

Les protocoles de routage non-sécuritaires ne prennent pas en compte l'égoïsme potentiel des participants. Lorsque ces protocoles sont utilisés au sein de réseaux militaires ou d'urgence, il est possible de prendre pour acquis que les différents nœuds collaboreront. Cette simplification ne peut toutefois pas être formulée pour les réseaux où l'accès est possible pour tous. Des usagers se souciant davantage de leur consommation énergétique que de la performance du réseau risquent donc d'utiliser le réseau

sans acheminer le trafic provenant des autres nœuds.La consommation de puissance étant un facteur déterminant, il faut donc à tout pris veiller à ce que les nœuds ne puissent agir de façon égoïste, tout en empêchant les attaquants de nuire.

Deux aspects se démarquent donc. Premièrement, il faut empêcher les nœuds malicieux de perturber le protocole de routage, en envoyant des messages de routage non conformes au protocole. Ensuite, il est nécessaire que chaque nœud achemine les paquets conformément à sa table de routage.

Différents mécanismes ont été développés. Les premiers, des mécanismes préventifs, veillent à ce que les tables de routage se créent de façon sûre en empêchant la contre-façon des différents paquets. Ils utilisent également la théorie des jeux afin de s'assurer que les nœuds n'auront pas d'avantages à contrevenir au protocole. Les seconds mécanismes, réactifs, détectent et excluent les nœuds fautifs. La réputation est un de ces mécanismes et est largement utilisée.

L'analyse de performance des contres-mesures utilisant la réputation (CMUR) est souvent faite en supposant que les attaquants ne tenteront pas de passer inaperçus. Cette simplification est très utile afin de simplifier le problème ainsi que les simulations. Par contre, il est fort peu probable que les attaquants ne tenteront pas de déjouer les mécanismes de sécurité. De plus, l'environnement utilisé lors de la modélisation est peu réaliste. Peu de collisions sont générées, ce qui facilite la détection des attaquants. Les indices de performances considérés sont peu pertinents et ne décrivent pas adéquatement l'interaction des attaquants avec leur environnement. Il est donc légitime de croire que la capacité de détection de ces mécanismes est moindre que ce que les simulations ont montré. Il est donc nécessaire de valider chacune des analyses de performance.

Toutefois, pour valider une analyse de performance, il est nécessaire de valider le simulateur utilisé, les choix de facteurs qui constituent le plan d'expérience ainsi que les paramètres expérimentaux. En effet, même si un simulateur commercial et reconnu est utilisé, il est souvent nécessaire de le modifier et d'implanter des fonctionnalités afin de les tester. Une erreur peut facilement se glisser et changer du tout au tout les conclusions qui peuvent être tirées d'une analyse de performance. Ensuite, ignorer certaines combinaisons de facteurs peut souvent avantager une CMUR. Ainsi, dans la validation de CONFIDANT, une CMUR décrite au chapitre suivant, peu de connections sont présentes dans les réseaux, ce qui diminue le nombre d'interférence et améliore le rendement des CMUR. Finalement, les paramètres expérimentaux, par

exemple la puissance d'émission des nœuds, influencent le déroulement de l'expérience et peuvent eux aussi changer les conclusions qu'une expérimentation peuvent apporter.

Il serait donc très utile d'avoir un modèle mathématique, indépendant de tout simulateur et language informatique, pour valider les CMUR et leur implémentation. En effet, il est souvent plus aisé de trouver une erreur dans une modélisation mathématique que dans un programme entier. Toutefois, aucun modèle analytique ne fait de relation entre l'interaction entre des attaquants, leur milieu et différents indices de performance.

1.3 Objectifs de recherche

L'objectif principal de ce mémoire est de créer un modèle analytique décrivant la relation entre l'environnement des RAHM, les attaquants et des indices de performances définis. Plus spécifiquement, les objectifs suivants sont visés :

- Analyser différents protocoles de routage sécuritaires utilisant le concept de réputation et définir les différentes attaques visant les RAHM.
- Définir un modèle décrivant la relation entre l'environnement des RAHM, les attaquants agissant dans les RAHM, les CMUR, la demande, le taux de perte de paquets réel et observé, le taux de faux positifs, le taux de faux négatifs et le taux de perte de paquets.
- Valider le modèle en le comparant avec un simulateur.

1.4 Plan du mémoire

Après cette introduction, quatre autres chapitres constituent ce mémoire.

- Le second chapitre est une revue de littérature. Les caractéristiques des RAHM y sont d'abord explicitées. Par la suite, les attaquants ainsi que les attaques visant les RAHM sont décrites. Suivent ensuite différents protocoles sécuritaires proactifs. Quelques éléments de la théorie des jeux sont introduits, suivie des protocoles utilisant la réputation. La revue de littérature se conclue avec une synthèse des problèmes qui restent ouverts.
- Le troisième chapitre est constitué de la description d'un modèle intégrant les caractéristiques des RAHM, les critères de performance appropriés et les

différentes caractéristiques des attaquants et des attaques qu'ils mettent en œuvre.

- La validation du modèle constitue le quatrième chapitre. Le modèle est implanté à l'aide du language C++ et le protocole DSR est simulé en présence d'attaque. Les résultats obtenus sont comparés à ceux obtenus à l'aide du simulateur Qualnet.
- Finalement, le cinquième et dernier chapitre est une conclusion qui fait la synthèse des différents concepts abordés dans le mémoire en insistant sur les contributions apportées par ce travail. Des pistes de travaux futurs se basant sur ces contributions sont apportées en exposant les leçons apprises lors de ce mémoire.

CHAPITRE 2

Attaques et mécanismes de prévention dans les RAHM

Avec la venue de la téléphonie mobile, les usagers ont pu utiliser différentes ressources à distance et en mouvement. Cette mobilité s'est peu à peu étendue aux réseaux informatiques, permettant ainsi aux utilisateurs de se déplacer tout en utilisant ces réseaux. Pour la plupart de ces réseaux, tant téléphoniques qu'informatiques, seul l'usager est mobile. Une infrastructure fixe est nécessaire pour qu'ils fonctionnent. Les RAHM qui n'utilisent aucune infrastructure fixe sont apparus dernièrement.

Un RAHM consiste en un ensemble de nœuds mobiles assumant les fonctions à la base du bon fonctionnement d'un réseau. Les nœuds effectuent donc le routage, l'acheminement des paquets et la découverte de services (*service discovery*), sans utiliser d'infrastructure dédiée telle que des points d'accès ou des stations de base. Les nœuds du RAHM acheminent les paquets des autres nœuds jusqu'à destination, ce qui leur permet de communiquer avec des stations situées à une distance supérieure à leur portée d'émission.Les nœuds peuvent être de différente nature, tels des ordinateurs portables, des assistants numériques portables (ANP) ou même des réseaux à part entière. Ils peuvent être en mouvement et peuvent entrer et sortir du réseau. Des exemples d'application de ces réseaux comprennent des réseaux militaires ou de réponse à un désastre, des activités extérieures ou encore des points de services situés dans des lieux publics.

Ces réseaux sont toutefois très vulnérables à plusieurs types d'attaques. Contrairement aux réseaux câblés, les RAHM partagent le médium de communication avec les adversaires potentiels et ne possèdent pas d'entité centrale qui permettrait de surveiller le réseau. Les protocoles de routage pour RAHM sont vulnérables à une panoplie d'attaques et des déni de services peuvent être effectués sans grande difficulté. La disponibilité de ces réseaux est donc grandement menacée, contrairement à la confidentialité des données et à leur intégrité, qui peuvent être assurées par des

moyens cryptographiques bien connus, telles les clefs asymétriques. Des mécanismes doivent être implémentés au niveau de la couche réseau afin d'assurer la disponibilité des RAHM.

Cette revue de littérature couvre donc différents aspects de la sécurité des RAHM et différents mécanismes visant d'abord et avant tout à améliorer la disponibilité de ces réseaux dans un milieu hostile. Tout d'abord, les caractéristiques générales des RAHM sont présentées afin de mieux cerner la problématique. Deuxièmement, les aspects de sécurité propres aux RAHM sont soulevés. Troisièmement, différents mécanismes de protection proactifs pour les RAHM sont décrits. Quatrièmement, la théorie des jeux telle qu'appliquée aux RAHM est introduite. Ensuite, les mécanismes utilisant la notion de réputation sont décrits. Ce chapitre se conclue avec une synthèse des problèmes ouverts.

2.1 Description des réseaux ad hoc mobiles

Comme décrit précédemment, un RAHM est un réseau où aucune infrastructure fixe, telle des points d'accès, n'est présente et où les différents nœuds participent au routage. Dans cette section, les caractéristiques des RAHM sont décrites plus précisément. Ensuite, les catégories de protocoles de routage sont énumérées. Finalement, les contraintes relatives à la sécurité sont introduites.

2.1.1 Caractéristiques des RAHM

Un groupe du *Internet Engineering Task Force* (IETF) a défini les caractéristiques précises d'un RAHM (Corson et Macker, 1999). Ces différentes caractéristiques, soit la présence de topologies dynamiques, les contraintes de bande passante et d'énergie, la sécurité physique restreinte et le vaste déploiement, doivent être prises en compte lorsque les différents protocoles sont développés.

Topologies dynamiques. Les nœuds sont libres de bouger arbitrairement. La topologie du réseau peut donc changer de façon aléatoire et rapidement. Les liens peuvent être aussi bien unidirectionnels que bidirectionnels. Il est important de noter que plusieurs protocoles de routage sécuritaire étudiés supposent malgré tout que les liens sont bidirectionnels.

Contraintes de bande passante. Les liens sans fils continuent d'avoir une capacité significativement inférieure aux liens câblés. De plus, les effets causés par les accès multiples, l'atténuation, le bruit et les interférences font que le taux de transmission réalisé est inférieur au taux de transmission théorique.

Un effet de cette contrainte est que la congestion constitue la norme plutôt que l'exception. La demande des nœuds approche ou dépasse régulièrement la capacité du réseau. Ceci est dû au fait que les RAHM sont l'extension des réseaux fixes et ont donc à assumer les mêmes services multimédias ou autres.

Contrainte d'énergie. Certains des nœuds fonctionnant dans un RAHM utilisent des batteries. Pour ces nœuds, conserver leur énergie est un critère parmi les plus importants. Cette caractéristique mène les nœuds à éviter de router le trafic qui ne leur est pas destiné pour conserver leur énergie plus longtemps. Plusieurs travaux ont étudié cette question (Anderegg et Eidenbenz, 2003; Hasswa *et al.*, 2005; Liu et Issarny, 2004; Marti *et al.*, 2000; Michiardi et Molva, 2002, 2003; Nurmi, 2004).

Sécurité physique restreinte. Les RAHM sont généralement plus vulnérables aux menaces physiques que les réseaux fixes. De plus, les possibilités d'écoute, de mascarade et de déni de service sont également accentuées. Par contre, la nature décentralisé du RAHM lui confère une robustesse supplémentaire en éliminant les points de défaillance uniques (*single points of failure*).

Vaste déploiement. Certains scénarios envisagés sont de grande taille, c'est-à-dire qu'ils contiennent plus d'une centaine de nœuds. Bien que le besoin de supporter un vaste déploiement ne soit pas unique aux RAHM, les mécanismes développés devront tenir compte de cette caractéristique. Par exemple, conserver la liste entière des nœuds et de leurs caractéristiques pourrait vite devenir trop lourd pour certains usagers.

2.1.2 Protocoles de routage

Plusieurs protocoles de routage pour RAHM ont été développés. Ces protocoles tiennent pour acquis que tous les nœuds agissent de façon conforme au protocole. Ils sont donc totalement vulnérables vis-à-vis n'importe quelle attaque présentée à la section 2.2. Même sans attaquants actifs dans le réseau, un nœud mal configuré peut entraîner le mauvais fonctionnement d'un réseau entier. Par contre, ces protocoles

sont très efficaces et sont à la base de la grande majorité des protocoles sécuritaires. Les protocoles de routage se séparent en deux grandes catégories, proactif ou réactif.

Protocole proactif. Un protocole proactif est un protocole où les nœuds échangent de façon périodique leurs tables de routage dans le but que chaque nœud ait une route pour chaque autre nœud.

Protocole réactif. Un protocole réactif, ou sur demande, est un protocole où les nœuds échangent les informations relatives au routage seulement lorsque cela est nécessaire. Ainsi, un nœud tente de trouver une route vers une destination uniquement s'il a un paquet à envoyer à cette destination. Ce type de protocole est souvent plus efficace qu'un protocole proactif. Il a un plus petit surdébit (*overhead*) et est capable de réagir plus rapidement aux changements de topologie qui sont fréquents dans les RAHM.

DSR

Le protocole Dynamic Source Request (Johnson *et al.*, 2004) est un exemple de protocole réactif. Lorsqu'un paquet doit être émis, la source émet un paquet *Route Request* vers tous les nœuds qui sont à sa portée. Cette requête comprend son adresse, celle de la destination et un numéro servant à identifier la requête. Ces nœuds envoient par la suite cette requête aux nœuds qui sont leurs voisins, en ajoutant leur adresse dans la requête. Si un nœud intermédiaire connaît une route vers la destination, il la fait parvenir au nœud source avec un paquet *Route Reply.* Quand la requête de route arrive à destination, elle renvoie un paquet *Route Reply* par les nœuds dont les adresses sont présentes dans la requête de route. Si d'autres requêtes ayant le même numéro d'identification et ayant transité par des chemins différents arrivent à destination, elles sont jetées. La même chose se produit si des requêtes arrivent à un nœud intermédiaire et que ce dernier a déjà reçu une requête avec le même numéro d'identification.

Lorsqu'un paquet *Route Reply* arrive à la source de la requête, elle ajoute cette route dans sa cache. Chacun des nœuds intermédiaires ajoutent également la route dans leur cache.

Si un nœud ne peut acheminer un paquet et qu'il a dans sa cache une autre route, il pourra envoyer le paquet par celle-ci. Sinon, il jettera le paquet. Dans les deux cas, il enverra un paquet *Route Error* à la source du paquet. Tous les nœuds intermédiaires

ainsi que le nœud source enlèveront la route de leur cache. Le nombre de requêtes de routes qu'un nœud peut effectuer est limité dans le temps afin que si un nœud est inaccessible, le réseau ne soit pas inondé de requêtes. Le temps entre les requêtes double entre chacun des essais successifs vers un même nœud.

2.1.3 Contraintes relatives à la sécurité des RAHM

Les RAHM sont généralement plus menacés que les réseaux fixes. Contrairement aux réseaux fixes, aucun routeur n'est présent. Cette fonction est assumée collectivement par chacun des nœuds. Ensuite, le médium de communication est accessible autant par l'utilisateur légitime que par l'attaquant. Il résulte de ces deux situations qu'il n'existe pas de ligne de défense claire dans les RAHM. Il n'y a plus d'infrastructure où une application implémentant la sécurité, telle un système de détection d'intrus (SDI) ou un coupe-feu, pourrait être déployée. De plus, les nœuds peuvent être compromis physiquement, compromettant ainsi le matériel cryptographique servant à l'authentification et permettant des attaques au niveau du protocole de routage.

Ces caractéristiques montrent l'importance de construire une solution de sécurité fonctionnant à l'aide de plusieurs lignes de défense qui assurerait un niveau de protection adéquat tout en maintenant un niveau de service acceptable. Cette solution doit être répartie chez tous les nœuds pour assurer la sécurité tant au niveau local que global. De plus, elle doit sécuriser les différentes couches du protocole, chaque couche fournissant sa ligne de défense. La solution doit repousser toutes les menaces, qu'elles viennent de l'intérieur du réseau ou de l'extérieur. Cette partie de la solution est décrite comme étant une approche préventive par certains auteurs (Yang *et al.*, 2004). Ensuite, le mécanisme de protection doit assurer la détection d'une attaque qui aurait contourné les mesures préventives afin d'enclencher des mécanismes de défense réactifs appropriés, tel d'exclure un nœud compromis du réseau. En effet, si une attaque réussit malgré la prévention, il doit être possible de la détecter afin d'empêcher l'attaquant de compromettre totalement le nœud attaqué et de corrompre la sécurité des autres nœuds du réseau. Finalement, la solution doit être pratique et doit respecter les contraintes en terme de ressources.

La conservation des ressources est elle aussi une problématique importante. Comme la capacité des nœuds en terme de calcul peut être faible, comme dans le cas des ANP, il est difficile d'implanter des solutions se fiant sur la cryptographie asymétrique.

De plus, la bande passante est réduite, rendant le réseau plus vulnérable aux attaques de déni de service basée sur des faiblesses propres aux protocoles sécuritaires. Une station forcée d'émettre plusieurs fois chaque message chiffré verra ses batteries se vider prématurément de même qu'un nœud forcé de vérifier des signatures asymétriques à répétition n'aura plus les ressources pour effectuer d'autres opérations.

2.2 Attaquants et attaques propres aux RAHM

Avant d'évaluer la sécurité d'un protocole de routage, il est nécessaire de décrire les différentes attaques que peuvent subir les RAHM et les différents types d'attaquants qui peuvent procéder à celles-ci. Les attaques décrites visent d'abord et avant tout à diminuer la disponibilité des ressources du réseau. La section qui suit présente premièrement les différents types d'attaquants, c'est-à-dire les attaquants actifs, passifs, internes ou externes. Ensuite, différentes attaques sont présentées, classées selon la couche de la pile de protocoles où elles opèrent. Des exemples concrets d'attaques visant la couche réseau sont donnés par la suite.

2.2.1 Catégories d'attaquants

Différentes catégories ont été crées afin de pouvoir classer les attaquants. Ces catégories se basent sur l'action qui est effectuée par l'attaquant ou sur la quantité de matériel cryptographique qui est en leur possession. Les attaquants sont tout d'abord séparés en deux groupes, les attaquants actifs ou passifs. Ces deux catégories sont décrites ci-dessous.

Attaquant actif. Dans leur revue de littérature respective, différents auteurs décrivent les attaques actives comme étant des attaques qui requièrent l'émission de la part de l'attaquant (Perrig et Hu, 2004; Molva et Michiardi, 2003). Un attaquant actif peut introduire des paquets dans le réseau afin de corrompre le protocole de routage comme il peut agir afin de limiter l'accès au médium de communication en émettant sans cesse.

Attaquant passif. Par contre, les mêmes auteurs divergent légèrement quant à la définition d'une attaque passive. Perrig et Hu considèrent qu'un attaquant passif ne fait qu'écouter sur le réseau. En plus de l'écoute, Molva et Michiardi associent à une attaque passive l'omission d'acheminer les paquets conformément à sa

table de routage dans le but d'économiser de l'énergie. C'est cette dernière définition d'attaquant passif qui sera utilisée tout au long de ce mémoire.

Il est très important de noter qu'il est à toute fin pratique impossible de prévenir une attaque passive. Cependant, différents mécanismes, tels l'utilisation de la réputation, peuvent punir les nœuds fautifs et éviter qu'ils sévissent à nouveau. Ce seront donc des mécanismes réactifs qui seront utilisés afin de réduire l'impact de ces attaques.

Les attaquants sont également classés en fonction du nombre de nœuds légitimes compromis et donc du matériel cryptographique possédé par l'attaquant. Un attaquant n'ayant pas accès au matériel cryptographique sera décrit comme un attaquant externe, dans le cas contraire comme un attaquant interne.

Attaquant externe. L'attaquant externe n'a aucun matériel cryptographique ou servant à l'authentification : son impact sera donc limité sur le réseau. Il ne pourra effectuer qu'un nombre limité d'attaques telles que l'écoute, le brouillage ou encore des attaques par répétition de paquets telles que décrites ci-dessous.

Attaquant interne. L'attaquant interne a acquis le matériel cryptographique et d'authentification nécessaire pour un au moins nœud. Il peut donc participer au protocole de routage. Son influence est donc beaucoup plus grande que s'il n'avait aucun matériel cryptographique en sa possession. Une ligue d'attaquants correspond à un groupe d'attaquants ayant le matériel cryptographique pour plusieurs nœuds et menant une attaque concertée.

2.2.2 Attaques contre les couches de la pile de protocoles pour RAHM

Les attaques sont généralement décrites en fonction de la couche de la pile de protocole qu'elles affectent. Cette classification est très utile si on veut mettre en place une stratégie de sécurité agissant à toutes les couches du RAHM. Les attaques présentées ci-dessous opèrent à la couche physique, à la couche liaison, à la couche réseau ou encore à la couche application de la pile de protocole.

Couche physique. Comme tout réseau utilisant une interface radio, une attaque peut être menée en brouillant le signal. Pour contrer cette menace, la couche physique doit utiliser un protocole utilisant un ratio signal sur bruit élevé.

Couche liaison. Ces attaques sont propres à chaque protocole de la couche liaison utilisé. Par exemple, de nombreuses vulnérabilités au sein du protocole 802.11 ont été identifiées afin d'effectuer des attaques par déni de service.Ces attaques utilisent la façon avec laquelle est gérée les accès au médium. En se servant des vulnérabilités des protocoles de la couche liaison, il est possible pour un nœud malicieux d'accaparer le médium et d'empêcher les nœuds qui l'entourent de communiquer.

Couche réseau. Au niveau de la couche réseau, les attaques peuvent être de deux types : une attaque contre le routage ou une attaque de consommation de ressources. Ces deux types d'attaques, du point de vue de l'utilisateur, peuvent être vues comme des attaques de déni de service.Des exemples concrets de ces types d'attaques sont décrits à la section 2.2.3.

Attaque contre le routage. Cette attaque consiste à empêcher le bon fonctionnement du protocole de routage. L'attaquant agira donc de façon non conforme au protocole afin de le perturber.

Pour un nœud, une façon simple de perturber le routage est de ne pas acheminer les paquets conformément à sa table de routage. Selon la définition utilisée dans ce mémoire, c'est une attaque passive.

Attaques de consommation de ressources. Cette attaque consiste à injecter des paquets non-conformes au protocole dans le réseau dans le but d'utiliser des ressources limitées telles que l'énergie, la bande passante, la mémoire, ou encore la puissance de calcul.

Couche application. Les vers (*worms*) représentent une grande menace envers les RAHM.Contrairement aux réseaux traditionnels où les serveurs, les usagers et les routeurs sont séparés et où les routeurs fournissent peu de services, les nœuds du RAHM sont à la fois routeurs, utilisateurs de services et serveurs. Une attaque au niveau application peut donc entraîner des répercussions au niveau du routage. De plus, la plupart des nœuds fonctionnent avec les mêmes logiciels et les mêmes systèmes d'exploitation. Une seule vulnérabilité pourrait donc permettre à un vers de se propager à l'ensemble du réseau plus rapidement qu'aucun mécanisme de détection ne pourrait détecter.

2.2.3 Attaques fréquentes contre le routage dans les RAHM

Voici plusieurs exemples d'attaques propres aux RAHM telles que décrites dans différents articles (Sanzgiri *et al.*, 2002; Zapata et Asokan, 2002; Perrig et Hu, 2004; Molva et Michiardi, 2003). La plupart de ces scénarios mettent en cause des attaques actives. Ce sont des attaques qui visent à corrompre les tables de routage des différents nœuds. Un protocole de routage sécuritaire devra empêcher les attaques actives de se produire. Voici les attaques les plus souvent énumérées dans la littérature.

Participation d'un nœud non autorisé. Dans cette attaque, un nœud utilise les ressources du réseau alors qu'il n'en a pas le droit.

Imposture. Un attaquant prend la place d'un nœud légitime et émet avec l'adresse IP ou MAC de ce nœud.

Fabrication de messages de routage. Un nœud envoie un message de routage qui n'est pas conforme au protocole. Par exemple, un nœud malicieux pourrait envoyer un *Route Error* alors qu'aucune erreur n'a eu lieu. Ceci peut être effectué dans le but d'accomplir une attaque de consommation de ressources.

Altération des messages de routage. Un nœud modifie un message de routage de façon non-conforme au protocole. Un nœud peut créer des routes sous-optimales en acheminant des paquets falsifiés. Ainsi, l'acheminement des paquets sera plus long. Aussi,des boucles peuvent être crées rendant ainsi impossible d'acheminer le trafic.

Attaque par répétition de paquets. Un nœud enregistre un paquet émis par un autre nœud et le renvoie tel quel.

Rushing attacks. Cette attaque vise particulièrement les protocoles de routage sur demande et est un cas particulier d'attaque par répétition de paquets. Cette attaque dissémine des *Route Request* forgés tout au long du réseau avant que les *Route Request* légitimes ne puisse se propager. Les *Route Request* légitimes seront donc jetés au profit des *Route Request* forgés, empêchant le protocole d'agir si ceux-ci sont corrompus et jetés par la destination.

Trous de ver. Une attaque particulièrement puissance et difficile à contrer est le trou de ver.C'est un autre cas particulier d'attaque par répétition de paquets. L'attaquant commence par enregistrer les paquets émis à la source. Ceux-ci sont acheminés à un second nœud malicieux via une connexion privée, par exemple

un lien câblé. Ces paquets sont par la suite réémis tels quels. Si tous les paquets sont acheminés de cette manière, l'attaquant rend service au réseau.C'est lorsque l'attaquant n'achemine que les paquets de contrôle qu'un problème surgit. En effet, la route générée par le trou de ver sera généralement beaucoup plus rapide et contiendra moins de sauts que les autres routes. Les nœuds auront donc tendance à vouloir utiliser cette route. Comme les paquets qui sont rejoués ne sont pas altérés, il sera très difficile de constater qu'il y a eu une attaque. Et lorsque cette attaque est bien effectuée, il sera même impossible de découvrir une route de longueur supérieur à deux sauts. De plus, l'utilisation de la cryptographie ne peut empêcher ce type d'attaque.

2.3 Mécanismes de prévention des attaques au niveau de la couche réseau

Cette section présente un aperçu des mécanismes de prévention des attaques au niveau de la couche réseau. Tout d'abord, les mécanismes de distribution des clefs et certificats sont présentés. Ces protocoles sont nécessaires au bon fonctionnement de la plupart des autres protocoles. Par la suite, des méthodes d'authentification sont décrites. Ces méthodes se fient sur les mécanismes de distribution de clefs et de certificats et sont inclus dans les différents protocoles de routage sécuritaires. Ensuite, un protocole de routage sécuritaire souvent cité dans la littérature, ARIADNE, est présenté (Hu *et al.*, 2002b). Des mécanismes de protection contre les trous de vers sont introduits (Hu *et al.*, 2003a), tout comme un protocole visant à augmenter la résistance des RAHM contre les *rushings attacks* (Hu *et al.*, 2003b). Finalement, la section se termine par une courte synthèse des mécanismes proactifs visant à assurer l'intégrité des tables de routage.

2.3.1 Distribution des clefs et des certificats

Les protocoles sécuritaires reposent en grande partie sur la distribution de certificats et de clefs. Il faut être capable d'associer une adresse IP à une et une seule clef pour communiquer avec le nœud. Lorsqu'une architecture à clef publique est utilisée, tous les nœuds doivent pouvoir prouver leur identité à l'aide d'un certificat. Tous les nœuds doivent donc avoir un certificat valide en tout temps.

Quatre grandes tendances résument les stratégies adoptées : la distribution des clefs à tous les nœuds préalablement à l'établissement du réseau, la présence d'une autorité centrale (qui peut être distribuée ou non), l'utilisation de répertoires de certificats comparables au modèle de l'application *Pretty Good Privacy* (PGP) et finalement les adresses semblables aux *Cryptographically Generated Adresses* (CGA).

Distribution préalable des certificats. Une des options est de distribuer les certificats à chaque nœuds avant qu'ils ne prennent part au réseau. La distribution préalable des clefs et certificats assume que le nombre d'usagers est connu à l'avance, ce qui peut mal s'appliquer à différents scénarios de RAHM.

Autorité centrale. Une autorité centrale peut veiller à maintenir les différents certificats à jour. Si les nœuds peuvent obtenir leur certificat avant d'entrer sur le réseau, l'autorité centrale assurera la fonction de maintenir à jour ces certificats et de les révoquer si un nœud ne respecte pas le protocole. Si un canal sécuritaire est disponible, il sera possible d'obtenir un certificat via ce canal pendant que le réseau est opérationnel. Un seul certificat doit être alors prédistribué, celui de l'autorité de certification.

Autorité centrale répartie. L'utilisation d'une autorité centrale a comme principal désavantage d'offrir un point de défaillance unique. Ainsi, un attaquant n'a besoin de prendre possession que de cette entité afin de compromettre tout un réseau. Aussi, une attaque par déni de service dirigée contre l'autorité et qui serait menée avec succès réussirait à empêcher le réseau de fonctionner.

Distribuer une autorité centrale sur plusieurs nœuds apporte comme avantage une plus grande résistance au déni de service mais rend la tâche de corrompre une autorité plus facile. Un attaquant peut simplement attaquer le nœud le plus faible et gagner la capacité d'émettre des certificats.

Des auteurs proposent d'utiliser la méthode nommée *threshold cryptography* (Zhou et Haas, 1999). Cette méthode permet à n parties d'émettre des certificats. De ces n nœuds, $m + 1$ doivent participer à la création de celui-ci, où $n > m$. Cette méthode crée donc une clef publique unique et une clef privée divisée en n parties. Pour que que le service signe un certificat, chaque serveur génère une signature partielle pour le certificat en utilisant sa clef privée et la soumet à un nœud capable de reconstituer une telle signature. Chaque serveur peut être un tel nœud et aucune information à propos des clefs privées ne lui est

donnée. Lorsque ce nœud spécial obtient $m + 1$ signatures valide, il est capable de générer la signature pour le certificat.

Tant que seulement m nœuds sont compromis, le système peut donc fonctionner. Ce système a donc comme avantage d'être résistant aux attaques de déni de service et oblige l'attaquant de compromettre plus d'un nœud. Par contre, il est nécessaire de distribuer le matériel cryptographique de façon sécuritaire aux différents serveurs avant de mettre le réseau en place.

Répertoires de certificats. Les répertoires de certificats sont utilisés afin de permettre aux usagers de procéder directement à l'authentification des autres usagers sans avoir à passer par une autorité centrale. Ce modèle est très semblable à celui utilisé par l'application PGP.

Des auteurs proposent donc le modèle suivant (Capkun *et al.*, 2003). Lorsqu'un usager en rencontre un autre, ils échangent leurs clefs et certificats respectifs via un lien sécuritaire comme un lien infra-rouge. Les usagers acquièrent les clefs et certificats au fur et à mesure qu'ils rencontrent les autres usagers. Lorsqu'un nœud doit communiquer avec un autre nœud, il commence par vérifier s'il a le certificat de ce nœud. Si tel est le cas, il peut communiquer directement avec lui après l'avoir authentifié. Sinon, les deux usagers comparent leur liste de certificats afin de trouver un tiers dont les usagers possèdent tous deux le certificat et qui a authentifié ces deux nœuds. Ainsi, il est possible de communiquer avec un autre usager sans l'avoir rencontré, à condition d'avoir un intermédiaire qui a la confiance des deux nœuds et qui a confiance envers ces deux derniers.

Par contre, les auteurs spécifient que ce mécanisme ne devrait s'effectuer qu'à travers un seul saut, c'est-à-dire que la propriété de transitivité ne peut être qu'à travers un et un seul nœud.

Cette façon de faire a plusieurs avantages, comme de minimiser le nombre de certificats à échanger préalablement à l'accès au RAHM. Par contre, les tests de performance effectués par les auteurs montrent que ce protocole est très inefficace, même si leurs auteurs tentent de dire le contraire. Ainsi, dans la plupart des cas de figure présentés, après 1000s, les nœuds auront moins de 50% des clefs et certificats des autres nœuds.

Cryptographically Generated Adresses. Tseng et Jiang présentent un mode de configuration des adresses IP propre à IPv6, protocole comprenant des adresses

de 128 bits. Il s'agit d'une technique qui construit une adresse IP à l'aide d'une clef publique.

Le nœud voulant intégrer le réseau commence par générer une clef publique k_p et une clef privée k_s. Il génère également un nombre aléatoire n_a ayant pour but d'éviter les collisions. À l'aide d'une fonction de hachage $H()$ décrite à la section 2.3.2, il génère les 64 bits derniers bits de son adresse locale. Ces bits sont égaux à $H(k_p, n_a)$. Les 64 premiers bits de son adresse peuvent servir à identifier le sous-réseau. Ainsi, chaque nœud associe de façon unique son adresse à sa clef publique.

Il est primordial de noter que ce mécanisme ne protège que l'adresse IP d'un nœud. En effet, il faudra utiliser un certificat approuvé par une autorité centrale ou un autre mécanisme afin d'authentifier de façon sécuritaire les applications des couches supérieures à la couche réseau et d'empêcher les attaques de type *man-in-the-middle*. Néanmoins, ce mécanisme est très utile dans le cadre d'un protocole de routage sécuritaire où aucun gestionnaire de certificat n'est présent.

2.3.2 Méhodes d'authentification symétriques

Tous les protocoles de routages sécuritaire utilisent un protocole d'authentification. Bien sûr, ce protocole peut être basé sur les clefs asymétriques, telles RSA. Mais leur utilisation demande une puissance de calcul qui peut être supérieure à celle offerte par les nœuds. Des protocoles visant à pallier à ce problème ont donc été présentés et sont décris ci-dessous. Le principal est Timed Efficient Stream Loss-tolerant Authentication (TESLA) (Perrig *et al.*, 2002) qui se base sur les chaînes de hachage. D'autres méthodes d'authentification utilisent les mêmes principes mais ne sont pas présentés dans cette revue de littérature.

Chaînes de hachage

Les chaînes de hachage sont un élément essentiel dans plusieurs publications portant sur la sécurité dans les RAHM (Yang *et al.*, 2004; Molva et Michiardi, 2003; Perrig et Hu, 2004; Zapata et Asokan, 2002; Papadimitratos et Hass, 2003; Perrig *et al.*, 2002; Tseng et Jiang, 2003; Hu *et al.*, 2002b,a). Ces chaînes permettent l'authentification d'un nœud ou d'un champ. De plus, elles sont faciles à calculer et peuvent être implantées même au sein d'unités ayant une très petite capacité de calcul.

Ces chaînes sont formées à l'aide d'une fonction de hachage cryptographiques H. Deux exemples de fonction de hachage couramment utilisées sont MD5 et SHA-1. Cette fonction cryptographique transforme une chaîne de bits de longueur quelconque en une chaîne de bits de longueur fixe. Si calculer $H(x)$ se fait très rapidement, calculer $H^{-1}(x)$ devrait être impossible dans un temps raisonnable. Voici une définition formelle de cette fonction.

$$H : \{0,1\}^* \mapsto \{0,1\}^\rho \tag{2.1}$$

où ρ est le nombre de bits de la chaîne produite par la fonction de hachage .

Pour créer une chaîne de hachage, un nœud choisit une chaîne de bits initiale au hasard, $x \in \{0,1\}^\rho$. Ensuite, les valeurs $h_0, h_1, h_2, h_3, ..., h_n$ où $h_0 = x$ et $h_i = H(h_{i-1})$ sont calculées. Pour la suite de ce chapitre, h_n est appelée l'ancre de la chaîne de hachage et x est la graine de la chaîne de hachage.

TESLA

Le protocole TESLA, proposé par Perrig *et al.*, est un protocole d'authentification permettant d'obtenir les caractéristiques de non-répudiation et d'authenticité nécessaires en utilisant seulement des primitives cryptographiques symétriques. L'idée sur laquelle se base TESLA est de créer un code d'authentification de message (CAM) à l'aide de clefs générées par des chaînes de hachage. Ces clefs sont relâchées selon un échéancier précis connu de tous et il est possible de vérifier l'authenticité de ces clefs. TESLA a comme contrainte que tous les nœuds qui participent à ce protocole doivent avoir des horloges synchronisées. Les nœuds doivent aussi pouvoir mettre en mémoire les messages reçus avant de pouvoir les authentifier. Nous décrivons ici les actions effectuées par un émetteur voulant signer un paquet et par le récepteur voulant vérifier cette signature.

Émetteur. Premièrement, l'émetteur du paquet doit séparer le temps en intervalles réguliers de grandeur T_{int}. Ensuite, il doit former une chaîne de hachage de longueur n ; $h_0, h_1, ..., h_{n-1}$. Il faut ensuite assigner les éléments de cette chaîne aux intervalles de temps dans l'ordre inverse de la génération.

L'émetteur distribue de façon sécuritaire (soit par un canal sécuritaire indépendant ou encore en signant le paquet à l'aide de clefs asymétriques) l'interval de temps T_{int}, le temps de départ T_0, la longueur de la chaîne de hachage N et

l'ancre de la chaîne de hachage h_n.

Utiliser les éléments de la chaîne de hachage pour générer les clefs et pour générer les autres éléments de la chaîne pourrait entraîner, selon les auteurs, des faiblesses sur le plan cryptographique. Une opération supplémentaire est donc effectuée sur un élément de la chaîne afin de générer cette clef. La fonction f est une fonction générant des nombres pseudo-aléatoires et résistant aux collisions. La fonction $F : F(i) = f_i(1)$ est utilisée pour calculer les clefs k_i associées avec chaque élément h_i de la chaîne de hachage : $k_i = F(h_i)$.

Voici les opérations que doit effectuer un émetteur à un temps t afin d'envoyer un message M. Pour l'intervalle de temps i, l'élément de la chaîne de hachage utilisé sera h_{n-i-1} et la clef qui lui correspond sera donc K_{n-i-1}

L'émetteur attache un CAM à chaque paquet. Ce CAM est calculé par rapport au contenu de chaque paquet à l'aide de la clef correspondant à l'intervalle de temps, dans ce cas K_{n-i-1}.

Le paquet envoyé sera donc

$$P = \{M||CAM(K_{n-i-1}, M)||K_{n-i-2}\}$$

ou $||$ est l'opérateur de concaténation.

Récepteur. Lorsque le récepteur reçoit le paquet, il le met en mémoire jusqu'à ce que l'émetteur envoie l'élément de hachage ayant servi à générer la clef. Lorsque le récepteur reçoit h_{n-i-1} , il vérifie que $H^{n-i-1}(h_{n-i-1}) = h_n$. Si tel est le cas, cet élément vient bel et bien de l'émetteur présumé. Le récepteur génère la clef K_{n-i-1} et vérifie si le CAM correspond bel et bien au paquet et a été généré avec la bonne clef. Si une de ces opérations est un échec, le paquet est jeté.

Comme TESLA n'utilise que des clefs symétriques, le temps de calcul pour authentifier un paquet est minimal. Il est ainsi possible de le faire en temps réel. Par contre, le désavantage principal de cet algorithme est que tous les nœuds doivent synchroniser, et ce de façon sécuritaire, leurs horloges. De plus, le délais d'attente servant à l'authentification peut augmenter le délais de bout-en-bout.

2.3.3 ARIADNE

ARIADNE (Hu *et al.*, 2002b) est un protocole de routage sur demande. Il est basé sur DSR et peut aussi bien utiliser comme protocole d'authentification TESLA que des signatures digitales comme RSA. ARIADNE se veut un protocole très efficace en terme de consommation de ressources et très sécuritaire.

Dans ce protocole, le paquet de requête de route (*Route Request*) est acheminé de la source à la destination. Chaque nœud qui achemine la requête ajoute son adresse au paquet et le signe. La réponse est réacheminée par les mêmes nœuds. Lorsqu'une route est hors-service, un paquet est envoyé afin de notifier les autres nœuds. Chaque nœud propageant ce message l'authentifie.

Afin de permettre son fonctionnement, l'échange de clefs et de certificats doit être effectué préalablement à la mise en marche du réseau. Les auteurs ne spécifient pas le type exact de mécanisme qui doit être effectué. Les liens reliant les nœuds doivent être bidirectionnels.

ARIADNE protège la formation des tables de routage en exigeant que chaque nœud signe le paquet qu'il fait parvenir au nœud suivant. Aussi, le champ correspondant au nombre de sauts du paquet est protégé en utilisant les chaînes de hachage. Ce champ est initialisé à une valeur h_0, valeur qui peut être déduite à partir de l'entête qui est elle-même authentifiée. À chaque saut, ce champ est ensuite haché à l'aide de la fonction de hachage utilisée sur l'ensemble du réseau. Ainsi, il est impossible de diminuer le nombre de saut et d'éliminer un nœud de la liste.

ARIADNE protège les RAHM contre la plupart des attaques à deux exceptions près. Les attaques par trous de vers peuvent être évitées grâce aux méthodes montrées à la section 2.3.4. Ces méthodes nécessitent toutefois une synchronisation très précise des horloges ou une connaissance de la position des différents usagers. Les attaques de type *rushing attacks* ne pouvant être empêchées à l'aide de ARIADNE, les mêmes auteurs ont développés certains mécanismes montrés à la section 2.3.5.

Malgré ces deux faiblesses, ce protocole de routage sécuritaire empêche la plupart des attaques de déni de service actives. Il n'empêche pas les attaques passives qui nécessitent des mécanismes de protection réactifs.

2.3.4 Protection contre les trous de vers

Afin de protéger les RAHM contre les trous de vers, des mécanismes de protection ont été introduits (Hu *et al.*, 2003a). Ces mécanismes veillent à ce qu'un paquet ne puisse voyager sur une distance qui serait supérieure à la portée normale d'un nœud. Ainsi, les paquets qui passent par un tunnel seraient détectés et jetés. Voici la notation utilisée par ces mécanismes :

- Δ : erreur maximale de synchronisation entre deux horloges
- c : vitesse de la lumière
- t_e : temps où le paquet expire
- t_s : temps où le paquet est émis
- L : distance maximale qu'un paquet peut parcourir
- L_{min} : valeur minimale pour L. $L > L_{min} = c\Delta$
- p_s : position de l'émetteur
- p_r : position du récepteur
- t_r : temps où le paquet est reçu
- δ : erreur maximale de positionnement d'un nœud
- ν : vitesse maximale d'un nœud

Ces mécanismes peuvent être classés en deux catégories :

Protection temporelle. À l'aide d'horloges synchronisées, il est possible d'établir le temps avant lequel un paquet arrivera à destination. Si un nœud reçoit un paquet et que ce dernier a parcouru une distance trop élevée, le temps d'arrivée du paquet sera supérieur à $t_e = t_s + \frac{L}{c} - \Delta$ et le paquet sera jeté.

La valeur de t_e est donc incluse dans le paquet à envoyer. Le paquet est ensuite signé selon le protocole d'authentification utilisé par le protocole de la couche réseau.

Le principal problème avec la protection temporelle est qu'un nœud ne sait jamais exactement quand il va envoyer un paquet. Ceci est dû au protocole de couche liaison. Ainsi, si le protocole 802.11 est utilisé, le nœud ne saura le moment où il émettra le paquet que $20\mu s$ à l'avance, soit le temps correspondant à l'émission d'une unité minimale de transmission (*minimum transmission unit*). Deux solutions sont proposées pour contourner ce problème. La première est d'utiliser un protocole d'authentification très efficace tandis que la seconde consiste à modifier le protocole de liaison de façon à accroître ce temps minimal

(Hu *et al.*, 2003a).

Protection géographique. Le principe derrière la protection géographique est lui aussi d'empêcher un paquet de parcourir une distance anormalement élevée. Ainsi, chaque paquet doit inclure dans le paquet sa position p_s, le temps où il émet le paquet t_s, et doit signer le paquet.

Lorsqu'un récepteur reçoit le paquet, il vérifie que

$$L > \| p_s - p_r \| + 2\nu * (t_r + t_s) + \delta$$

c'est-à-dire si la distance parcourue par le paquet est inférieure à la distance maximale. Si oui, le paquet est accepté, sinon, il est jeté.

En certaines circonstances, il serait possible d'effectuer une attaque par trou de vers si la protection géographique est utilisée. En effet, certains obstacles géographiques peuvent empêcher les transmissions. Un attaquant pourrait alors servir de relais entre l'émetteur et le récepteur et effectuer un trou de vers. Pour éliminer cette possibilité, chaque nœud posséderait un modèle de propagation des ondes. Les nœuds pourraient donc vérifier si une communication directe entre eux est possible.

Ces méthodes ont de grands désavantages : elles requièrent un équipement spécialisé et une synchronisation plus ou moins grande des horloges. Ces solutions ne sont donc pas toujours applicables dans des scénarios réels. De plus, il est nécessaire de modifier les protocoles afin de prévoir quel est le moment où le paquet sera envoyé.

De plus, la protection géographique veille à ce que chaque nœud dévoile sa position. Ceci pourrait aider un usager malveillant à déterminer la position exacte d'un usager et pourrait compromettre la sécurité physique de ce dernier. En effet, certains utilisateurs n'aimeraient pas dévoiler leur position.

Finalement, établir et maintenir un modèle de propagation des ondes n'est pas toujours applicable dans le cas d'un RAHM.

2.3.5 Protection contre les *rushing attacks*

En analysant le fonctionnement de la plupart des protocoles de routage sécuritaire, il est facile de constater qu'aucun de ceux-ci ne protège contre les *rushing attacks*. Aucun des protocoles de routage sécuritaire pour RAHM actuel ne protège contre

cette catégorie d'attaques (Hu *et al.*, 2003b). Les auteurs proposent toutefois un mécanisme pouvant apporter cette protection. Rushing Attack Prevention (RAP) est un mécanisme de protection pouvant être ajouté à de nombreux protocoles de routage sécuritaire, entre autre ARIADNE.

Ce protocole prévient les *rushing attacks* avec une probabilité de $(\frac{n-m}{n})^l$ où n est le nombre de routes entre la source et la destination (légitimes ou illégitimes), m est le nombre de nœuds illégitimes voisins de l'émetteur et l est le nombre de sauts entre la source et la destination. Sous attaque, ARIADNE et DSR ne parviennent pas à trouver aucune route alors que RAP trouvera une route après un certain nombre d'essais. De plus, il peut être intégré à de nombreux autres protocoles de routage pour RAHM afin de les protéger de ces attaques.

Par contre, ce protocole nécessite lui aussi un mécanisme de synchronisation des horloges. De plus, aucun lien unidirectionnel ne peut être utilisé. Ensuite, le protocole de couche liaison ajoute souvent un délai qui peut rendre RAP incapable de trouver des voisins. Un autre problème est que même avec un trafic très léger, RAP entraîne une importante dégradation du service. Alors qu'avec un certain trafic, ARIADNE et DSR arrivent à acheminer la presque totalité du trafic, RAP n'arrive qu'à en acheminer la moitié. De plus, RAP impose un surdébit beaucoup plus grand que ARIADNE ou DSR. C'est pourquoi les auteurs suggèrent de n'utiliser RAP qu'en cas de besoin.

RAP fonctionne en trois étapes. La première consiste à détecter de façon sécuritaire les voisins, la deuxième est la délégation sécuritaire de route et la troisième est l'acheminement randomisé des requêtes. Ces trois étapes assurent au protocole qu'il est possible de trouver une route valide selon une certaine probabilité. Voici ces trois étapes.

Détection sécuritaire des voisins. Cette étape consiste à vérifier qu'un nœud est bel et bien son voisin. La méthode utilisée est d'envoyer un premier paquet en broadcast (peut aussi être en unicast) un *neighbor sollicitation*. Cette requête contient un nombre aléatoire identifiant la requête et est signée par l'émetteur. Lorsqu'un voisin reçoit cette requête, il répond en incluant un nombre aléatoire à la suite du premier, signe le paquet et renvoie le paquet au premier nœud. Le premier nœud prend ce paquet, y ajoute à nouveau sa signature et le renvoie. Ainsi, chaque nœud est capable de calculer la distance maximale le séparant d'un voisin. Cette distance est égale à $\frac{\Delta c}{2}$ où c est la vitesse de la lumière et Δ

est le temps entre l'envoi du paquet et la réponse qui y correspond. Évidemment, la signature du paquet doit être très rapide tout comme le protocole de la couche liaison.

Délégation sécuritaire de route. Ce mécanisme vérifie que tous les nœuds qui participent à la délégation sécuritaire ont vérifié que leurs voisins supposés sont bel et bien à l'intérieur d'une certaine distance.

Acheminement randomisé des requêtes. La dernière étape consiste à attendre la réception de n instances d'une même requête, n étant choisi au hasard. Lorsque ces requêtes ont été reçues, une requête est choisie parmis les n.

2.3.6 Synthèse des mécanismes de défense proactifs

Beaucoup d'autres protocoles visant à assurer la convergence des tables de routage en présence d'attaquants ont été développés. Ils ont tous leurs forces et leurs faiblesses. Quant à ceux présentés dans cette section, ils assurent aux réseaux l'intégrité de leurs tables de routage tout en limitant les risques de déni de service et en assurant la propriété de non-répudiabilité. Bien sûr, certains protocoles sont difficilement applicables : les méthodes empêchant les trous de vers demandent d'incorporer des GPS au sein des nœuds tandis que RAP accroît sensiblement le surdébit. Il est toutefois possible de prendre pour acquis que les nœuds légitimes d'un RAHM peuvent créer des tables de routages fonctionnelles malgré la présence d'attaquants. Cette hypothèse sera utilisée lors de la conception du modèle au chapitre 3. Toutefois, ces mécanismes ne peuvent pas empêcher les attaques passives.

2.4 Théorie des jeux appliquée aux RAHM

Aucun mécanisme préventif ne peut à lui seul empêcher les nœuds malicieux de laisser tomber les paquets (Yang *et al.*, 2004). Il leur sera toujours possible de participer à la création des tables de routage et par la suite de ne pas s'y conformer. Comme les nœuds ont une batterie ayant une quantité d'énergie restreinte, il leur sera plus avantageux de laisser tomber les paquets qui ne leurs sont pas destinés à moins qu'on ne les force à acheminer le trafic. Si tous les nœuds suivent cette stratégie, aucun nœud n'acheminera de paquet et personne ne pourra donc communiquer. Une façon

de contraindre les nœuds de collaborer est d'appliquer les principes de la théorie des jeux.

La théorie des jeux est un domaine de connaissances ayant pour but d'étudier les interactions entre des usagers agissant dans leur propre intérêt et de modéliser leur stratégie de choix. Un jeu est un ensemble de règles qui définissent les différentes actions qu'un participant peu prendre. Dans ce cadre, chaque participant possède une fonction d'utilité qu'il tente de maximiser, ce qui constitue le but du jeu. Cette fonction représente ce qu'il peut gagner ou perdre dans ce jeu. Dans le cas d'un RAHM, cette fonction calcule le niveau de satisfaction qu'obtient un nœud à partir de l'utilisation des ressources du réseau, le niveau d'énergie dans sa batterie et l'utilisation de la bande passante (Michiardi et Molva, 2003). Les joueurs suivront donc une stratégie qui maximisera leur fonction d'utilité personnelle jusqu'à ce qu'un équilibre soit atteint.

Le concept d'équilibre est décrit comme l'entente optimale entre les différents joueurs d'un jeu. Dans une situation d'équilibre, les actions des joueurs sont telles qu'aucun joueur ne peut faire augmenter sa fonction d'utilité en changeant sa stratégie. Lorsqu'un jeu s'établit à un équilibre, il est possible que cet équilibre ne soit pas le meilleur pour l'ensemble des nœuds vu qu'ils agissent de façon égoïste. Aucun nœud ne peut augmenter sa fonction d'utilité en modifiant sa stratégie si les autres nœuds refusent de collaborer avec lui. Par contre, si tous les nœuds adoptent un comportement visant à maximiser la fonction d'utilité moyenne, ils pourraient faire augmenter du même coup leur propre fonction d'utilité.

Dans le cas des RAHM, les règles et les fonctions d'utilité des nœuds peuvent être déduites à partir des différents protocoles. La définition d'un jeu représentant un RAHM sécuritaire a pour but que la stratégie prise par chaque joueur indépendant s'accorde avec le but désiré par le concepteur du jeu, communément appelé l'optimum global (Anderegg et Eidenbenz, 2003). En autre mots, le jeu doit être conçu de façon à ce que la stratégie qui fait parvenir la situation à l'optimum social soit une stratégie que chaque joueur a intérêt de suivre.

Divers auteurs se basent sur cette théorie afin de développer des protocoles qui décourageraient l'égoïsme des différents usagers. L'utilisation d'une monnaie au sein d'un RAHM est une méthode qui a été développée (Buttyàn et Hubaux, 2001). Une autre façon d'aborder le problème est d'utiliser la réputation des différents nœuds pour déterminer la meilleure route à utiliser. Des protocoles utilisant la réputation

sont étudiés à la section 2.5.

2.4.1 Nuglets

Dans le but d'inciter les nœuds d'un réseau à faire un usage modéré des ressources et à acheminer le trafic, Buttyàn et Hubaux proposent d'utiliser une monnaie, le *nuglet*. Cette monnaie serait échangée entre chaque participants lors de l'acheminement des paquets. Le principe est le suivant : chaque fois qu'un nœud envoie ou reçoit des paquets, il doit dépenser des *nuglets*. Chaque fois qu'il achemine des paquets, il reçoit des *nuglets*. Comme les nœuds doivent dépenser lorsqu'ils émettent pour leur propre compte, ils devront dépenser pour effectuer un déni de service ce qui devient très rapidement coûteux. Aussi, pour ne pas se retrouver sans *nuglets* et ainsi être incapable de communiquer, ils devront nécessairement acheminer des paquets. Deux modèles de fonctionnement sont proposés par les auteurs, le Paquet Purse Model et le Paquet Trade Model. Ces deux variations sont présentées ci-dessous. Par la suite, les faiblesses des *nuglets* sont présentées.

Paquet Purse Model. Dans ce modèle, l'émetteur charge le paquet de *nuglets*. Chaque nœud prend un nombre défini de *nuglets* lorsqu'il achemine le paquet conformément au protocole de routage. Ce protocole a comme désavantage qu'il faut connaître la dimension du réseau afin de mettre le bon nombre de *nuglets* dans le paquet. Si la source n'en met pas assez, le paquet sera perdu, tout comme l'investissement. Si la source en met trop, l'excédant sera lui aussi perdu. Les auteurs suggèrent d'utiliser l'excédant de *nuglets* afin de payer des services offerts en ligne.

Paquet Trade Model. Dans ce modèle, l'émetteur vend le paquet au plus offrant. Ensuite, le nœud qui a acheté le paquet revend au plus offrant le paquet et ainsi de suite. Le coût de l'acheminement est donc assuré par la destination du paquet. Ce protocole a comme désavantage qu'il n'empêche pas un nœud d'inonder le réseau de paquets. Par contre, il n'est pas nécessaire de connaître à l'avance le nombre de sauts entre l'origine et la destination.

Le problème avec ce protocole est qu'une attaque par consommation de ressources est facilement envisageable. Lorsqu'un paquet est perdu, les *nuglets* qui s'y trouvaient sont perdus eux aussi comme le mentionnent d'ailleurs Buttyàn et Hubaux. Il serait

donc très aisé de ruiner un nœud. Lorsqu'un nœud émet un paquet chargé de *nuglets*, l'attaquant n'aurait qu'à laisser celui-ci communiquer avec un nœud adjacent et empêcher ce dernier nœud d'acheminer le paquet. Si le Paquet Trade Model est utilisé, le nœud adjacent aura perdu son investissement tandis que si le Paquet Purse Model est utilisé, le nœud source aura perdu tous les *nuglets* mis dans le paquet.

2.5 Réputation dans les réseaux

Une façon d'obtenir un jeu ayant un optimum social comme stratégie dominante pour chaque joueur est d'acheminer le trafic selon la réputation de chaque nœud. Ainsi, si un nœud veut que son trafic soit acheminé, il devra faire transiger une proportion minimale du trafic provenant des autres nœuds. Plusieurs protocoles se fient sur la réputation avant de choisir quelle sera la route utilisée pour acheminer le trafic. D'autres protocoles exclueront du réseau les usagers ayant une trop mauvaise réputation.

CORE et CONFIDANT utilisent la réputation afin de déterminer quelle est la meilleure route à utiliser pour acheminer le trafic et afin de déterminer les actions à prendre envers les nœuds malveillants. Ces mécanismes, très souvent cités dans la littérature, sont donc présentés. Ensuite, le mécanisme *Watchdog* (Marti *et al.*, 2000), qui utilise la réputation des nœuds afin de déterminer la route pour les paquets, est exposé. Routeguard, qui fonctionne selon le même principe est introduit par la suite. Finalement, différentes méthodes de calcul de réputation sont présentées. Elles sont plus résistantes aux différentes attaques qui visent les systèmes utilisant les réputations.

2.5.1 CORE

CORE (Michiardi et Molva, 2002), est un protocole qui traite le problème des nœuds malicieux ainsi que celui des nœuds égoïstes. Chaque nœud tient un registre contenant des métriques quantifiant la réputation des autres nœuds. La réputation peut prendre des valeurs allant de -1 (nœud complètement malicieux) à 1 (nœud totalement conforme au protocole). Si ces nœuds ont agit conformément au protocole de routage et à celui d'acheminement de paquets, leur réputation sera peu à peu augmentée. Sinon, leur réputation diminuera et mènera à leur exclusion du réseau

lorsqu'elle négative. Un nœud inconnu obtient une réputation initiale de zéro. La réputation des différents nœuds est échangée entre eux afin qu'un nœud puisse se dresser un portrait global du RAHM.

Lorsque la réputation d'un nœud est positive, cette réputation diminue progressivement avec le temps. Ce mécanisme empêche un nœud de participer au réseau seulement le temps de faire acheminer ses paquets pour ensuite retomber en veille.

Les auteurs notent que CORE est résistant aux attaques de déni de service qui pourraient être menées en envoyant à répétition des rapports négatifs à propos d'un nœud vu que seules les réputations positives sont transmises entre les nœuds.

Le protocole CORE est caractérisé par sa façon de calculer le routage et par les entités fonctionnelles comprises par chaque nœud.

Réputations. Trois types de réputations ont été définies par les auteurs. Ces trois types de réputations sont pondérées et combinées afin de former la réputation globale d'un nœud.

Réputation subjective. La réputation subjective est celle qui est calculée directement par un nœud à partir de ses observations. Cette réputation prend compte des événements qui se passent à un temps t ainsi que toutes les observations passées. Un poids plus important est accordé aux événements passés. Ainsi, un manquement sporadique au protocole aura peu d'effet sur la réputation d'un nœud. Cette façon de faire entraînera moins de faux positifs causés par le mauvais fonctionnement d'un lien ou aux collisions. Lorsque trop peu d'événements sont survenus, une réputation nulle est accordée au nœud.

Réputation indirecte. La réputation indirecte est la partie de la réputation qui est fournie par les autres nœuds. Cette réputation ne peut être que positive. Ainsi, un nœud voulant dégrader la performance d'un autre ne pourra pas le faire en envoyant une réputation négative qui serait fausse.

Cette réputation est calculée lorsque les nœuds acheminent les paquets. Si le paquet se rend à destination, la réputation des nœuds impliqués augmentera. La liste des nœuds ayant collaboré au protocole est incluse dans le paquet de retour.

Réputation fonctionnelle. La réputation fonctionnelle permet de calculer la réputation subjective et indirecte en fonction de différents critères ou observations. Ainsi, il est possible de calculer la réputation subjective d'un nœud relative à l'acheminement de paquets ou par rapport à sa conformité au protocole de routage.

Entités fonctionnelles. CORE définit deux types d'entités fonctionnelles : la table de réputations et le module de surveillance (*watchdog*).

Table de réputations. La table de réputations est une structure de données qui est contenue par chaque entité du réseau. La table des réputations contient la réputation de tous les autres entitées. Quatre entrées sont présentes.

1. Un identificateur unique identifiant chaque nœud
2. Un ensemble d'observations récentes faites sur le comportement de chaque entité
3. Une liste des réputations indirectes fournies par les autres entités
4. La réputation évaluée pour une opération prédéfinie, c'est-à-dire la liste des réputations fonctionnelles.

Chaque entité a une table de réputations pour chaque opération qui doit être surveillée. Aussi, une table de réputations globale est présente afin de combiner les différentes valeurs fournies par les autres tables.

Module de surveillance. Le module de surveillance est l'entité qui détecte le bon ou le mauvais fonctionnement des autres nœuds. Chaque fois qu'un nœud doit vérifier la validité d'une opération effectuée par un nœud voisin, il déclenche le module de surveillance spécifique à cette fonction. Celui-ci connaît l'état final correct de l'opération concernée. Si l'opération est correctement effectuée, le résultat de celle-ci concordera avec le résultat présumé ; le module de surveillance retombera alors en mode inactif. Sinon, une valeur négative est apportée dans la table de réputation concernée par l'opération observée.

Ce protocole comprend certaines limites. Premièrement, CORE prend pour acquis que tous les liens sont bidirectionnels. Certains RAHM ne peuvent donc pas utiliser CORE. Ensuite, si beaucoup de collisions sont présentes sur le réseau, le nombre de

faux positifs devrait être très élevé. En effet, il est impossible de déterminer si un paquet a été jeté à cause des collisions ou parce que le nœud est malicieux.

2.5.2 CONFIDANT

CONFIDANT (Buchenner et Le Boudec, 2002a) est une méthode de protection des nœuds qui fait appel à la détection des nœuds malicieux et à une réaction contre ces nœuds. Il propage la réputation à l'aide de messages ALARM.

Le protocole implante quatre mécanismes qui sont présents dans chaque nœuds.

Moniteur. Les nœuds qui sont les plus suceptibles de détecter un nœud malicieux sont dans le voisinage immédiat de ce dernier. Les mauvais comportements suivant peuvent être détectés :

- Aucun acheminement des paquets.
- Attraction du trafic hors de la normale, c'est-à-dire que le nœud indique de très bonnes routes ou indique des routes très rapidement de façon à ce qu'il achemine le plus de trafic possible.
- Réacheminer le routage sans qu'aucune erreur n'ait eu lieu.
- Ne pas envoyer de messages d'erreur même si une erreur a été détectée.
- Mise à jour des routes à une fréquence anormalement élevée.
- Acheminement des paquets de façon non conforme au protocole de routage.

Le moniteur enregistre ces déviations du comportement normal et dès qu'un comportement non-conforme est identifié, le système de gestion des réputations est appelé.

Gestionnaire de confiance. Ce composant est chargé de gérer les messages ALARM entrants et sortants venant aussi bien du moniteur du nœud que des autres nœuds. Ces messages indiquent les nœuds fautifs ainsi que la faute qui leur est reprochée. Le gestionnaire de confiance a pour principales fonctions :

- Calculer les niveaux de confiance envers les autres nœuds.
- Gérer la table contenant toutes les entrées contenant le niveau de confiance relatif aux nœuds.
- Acheminer les messages ALARM.
- Filtrer les messages ALARM conformément au niveau de confiance relafif au nœud d'où émane le message ALARM.

De plus, le gestionnaire de confiance est composé des éléments suivant :

- Une table contenant des informations concernant tous les messages ALARM reçus.
- Une table contenant les niveaux de confiance relatifs aux nœuds.
- La liste des nœuds jugés honnêtes à qui les messages ALARM sont envoyés.

Système de gestion des réputations. Le système de gestion des réputations gère une table consistant des différents nœuds et de leur réputation respective. Il gère cette table en distinguant les fausses accusations des nœuds réelement malicieux. La façon de faire qui réduirait le nombre de faux positifs n'est toutefois pas élaborée en détails par les auteurs.

Gestionnaire de routes. Ce composant est la partie qui réagit aux différentes observations du système. Les différentes fonctions sont :

- Trier les chemins selon des métriques ayant trait à la sécurité.
- Détruire les chemins contenant des nœuds malicieux.
- Agir sur la réception d'une requête ayant transité par un nœud malicieux.
- Agir sur la réception d'une requête ayant été émise par un nœud malicieux.

Les tests sur GloMoSim montrent que sous les attaques effectuées, CONFIDANT arrive à faire fonctionner DSR même si 60% des nœuds sont malicieux. Comme pour CORE, il est probable que le taux de faux positifs augmente avec le taux de collisions. Par contre, l'analyse de performance semble montrer le contraire. En effet, les figures 3 et 6 de l'article (Buchenner et Le Boudec, 2002b) montre que le taux de pertes de paquets est stable en fonction du nombre de nœuds et de connections au niveau applications. Cette analyse de performance est toutefois facilement contestable. Pour commencer, la portée des nœuds est très grande, c'est-à-dire de 250 m. Il y a donc de très nombreuses communications qui n'auront pas besoin de retransmissions. De plus, le nombre de connections simultanées est de 10 dans les simulations à 10 nœuds et 20 nœuds. Il aurait été intéressant de voir plus de connections afin de voir varier la performance du réseau lorsque des collisions sont présentes.

De plus, les paquets de données ne comprennent que 64 octets de données, ce qui est très petit comme taille de paquet. L'analyse de performance ne mentionne d'ailleurs pas la capacité du canal. Ainsi, si la vitesse utilisée est de 55 Mbps, il est normal que le taux de congestion soit relativement faible

En résumé, cette analyse de performance ne permet pas de savoir si la CMUR est efficace lorsque le réseau est congestionné, étant donné que le nombre de connections est trop faible.

2.5.3 Watchdog

La technique proposée par Marti *et al.* est d'utiliser la réputation des nœuds afin de définir quelle est la meilleure route à utiliser afin d'acheminer le trafic. Comme les nœuds égoïstes ne sont pas mis à contribution, ils ne peuvent pas jeter les paquets et ainsi faire augmenter le taux de perte de paquets. Par contre, ce mécanisme n'empêche pas les nœuds d'être égoïstes. Au contraire, ils sont avantagés. Ainsi, ils peuvent faire parvenir leur trafic à destination sans problème et les autres nœuds évitent de faire transiter leurs paquets par eux (Hasswa *et al.*, 2005).

Les principaux mécanismes proposés par les auteurs sont décrits ci-dessous en commençant par le *watchdog*. Ce mécanisme détermine la réputation des différents nœuds. Par la suite, le gestionnaire de route est décrit. Ce mécanisme détermine quelle est la meilleure route pour acheminer les différents paquets.

Watchdog. Ce mécanisme fonctionne en gardant en mémoire les paquets récemment envoyés à un nœud. Il les compare aux paquets envoyés par ce nœud. Si les paquets concordent, ils sont retirés de la mémoire. Si après un certain temps un paquet en question n'a pas été retransmis, le *watchdog* augmente un compteur. Lorsque ce compteur dépasse un certain seuil, le nœud est déclaré comme étant malicieux. Un message est envoyé à la source du paquet afin de l'avertir de cette situation.

Cette méthode a comme limite que, comme pour la majorité des méthodes utilisant les réputations, il est nécessaire que les liens soient bidirectionnels. Aussi, la probabilité de faux positifs devrait être fortement influencée par la quantité de collisions. L'analyse de performance faite par les auteurs ne vérifie toutefois pas cet impact.

Gestionnaire de route (pathrater). Le *pathrater* choisit la route que devront prendre les paquets afin d'être acheminés. Il assigne à chaque nœud avec qui il communique une réputation. Cette réputation vaut au maximum $1,0$, et ce seulement dans le cas du nœud qui observe. Sinon, cette réputation peut atteindre $0,8$ au maximum. Dans le cas d'un nœud qui est identifié par le *watchdog* comme étant malveillant, cette réputation prend la valeur -100. Lorsqu'un nœud est rencontré pour la première fois, sa réputation est égale à $0,5$. Lorsque le *pathrater* choisit la route, il trouve la route où les nœuds ont en moyenne la meilleure réputation.

Lorsqu'un paquet est acheminé par une route et que le paquet se rend à destination, tous les nœuds voient leur réputation augmentée de $0,01$. Si le paquet n'arrive pas à destination, leur réputation est diminuée de $0,05$. Lorsqu'un nœud a une réputation négative, elle est augmentée tranquillement ou encore remise à une valeur non-négative après un certain temps. Toutefois, cette procédure n'a pas été testée par les auteurs.

2.5.4 Routeguard

Routeguard est une CMUR a été récemment proposée (Hasswa *et al.*, 2005). Des mécanismes très proches de ceux proposés par Marti *et al.* sont utilisés. Un *watchdog* détecte les nœuds qui laissent tomber le trafic au lieu de l'acheminer. Par la suite, un mécanisme de routage ajuste les routes afin d'éviter les nœuds problématiques et de les exclure du réseau. Dans cette sous-section, le fonctionnement de Routeguard est présenté. Ensuite, les différentes faiblesses de Routeguard sont exposées.

Fonctionnement de Routeguard

Routeguard fonctionne sur tous les nœuds du réseau. Ceux-ci vérifient si leurs voisins acheminent correctement les paquets afin d'exclure les délinquants. Ainsi, chaque nœud garde une liste de réputation pour chacun de ses voisins. Cette liste lui est propre, Routeguard n'incluant aucun mécanisme de partage de la réputation.

Comme décrit précédemment, un module de surveillance repère les nœuds qui n'acheminent pas les paquets conformément à leur protocole de routage. Selon leur niveau de coopération, les nœuds sont classés en cinq catégories : nouveaux (*fresh*), membres (*member*), instables (*unstable*), suspects (*suspects*) ou malicieux (*malicious*). À chacune de ces catégories correspond un état. Ces états déterminent à leur tour le niveau de privilège qu'auront les nœuds du réseau.

Lorsqu'un nœud entre dans le réseau, il est dans l'état nouveau et a une réputation de zéro. Dans cet état, il n'a pas le droit d'envoyer de paquets mais doit acheminer les paquets des autres. Pour chaque paquet correctement acheminé, l'expéditeur augmente la réputation du nœud de un. Pour chaque paquet jeté, il diminue sa réputation de quatre. Si la réputation du nouveau nœud est positive après un certain temps t_n, il passe à l'état membre, sinon il passe à l'état suspect.

Lorsqu'un nœud est membre du réseau, il peut envoyer des paquets aux autres

nœuds afin de les faire parvenir à destination. Il doit aussi acheminer les paquets provenant des nœuds membres du réseau. Pour chaque paquet acheminé, la réputation d'un nœud est augmentée de un. Pour chaque paquet jeté, sa réputation est diminuée de cinq. La réputation peut atteindre un maximum noté $Tmem_{max}$. Si la réputation d'un nœud membre descend sous une certaine valeur, $Tmem_{min}$ le nœud passe à l'état instable.

Lorsque le nœud est jugé instable, il n'a plus le droit d'envoyer ses paquets sur le réseau et doit acheminer les paquets des autres nœuds. Sa réputation est réinitialisée à zéro. Si le nœud achemine un paquet, sa réputation est augmentée de un. Sinon, elle est diminuée de six. Si, au bout d'un certain temps t_u, la réputation de ce nœud est supérieure à zéro, il retourne à l'état membre. Sinon, il devient un nœud suspect. Un compteur, $Malcount$, est incrémenté à chaque fois qu'un nœud devient instable. Si ce compteur dépasse une valeur prédéfinie, il devient suspect.

L'état suspect est un état d'isolement. Le nœud est totalement isolé du réseau pour une période p. Après ce temps, il est reconnecté au réseau pour une période $0,5p$. Si le nœud se comporte de façon conforme pendant ce temps, il peut passer à l'état instable et son compteur $Malcount$ est remis à zéro. Sinon, il est classifié comme malicieux et ne peut plus se connecter au réseau. Il est banni à jamais du réseau et son adresse est gardée dans une liste servant à cet effet.

Faiblesses du protocole Routeguard

Les auteurs notent que le protocole fonctionne beaucoup mieux lorsque les nœuds malicieux laissent tomber tous les paquets plutôt que s'ils n'en laissent tomber qu'une partie. Selon eux, le pire scénario se produit lorsque les nœuds malicieux tentent de déjouer Routeguard. Pourtant, les résultats présentés ont été obtenus à l'aide d'un scénario plus avantageux.

En analysant le protocole, il est possible de voir qu'un attaquant qui, une fois parvenu à l'état membre, peut laisser tomber une partie du trafic tout en demeurant dans cet état. Une fois que le nœud obtient la valeur $Tmem_{max}$, il peut laisser tomber 1 paquet sur 6 tout gardant sa réputation égale à $Tmem_{max}$. Lorsque le nœud malicieux laisse tomber un paquet, sa réputation descend à $Tmem_{max} - 5$. Par la suite, en acheminant les 5 autres paquets, sa réputation redeviendra intacte. Il serait donc intéressant de vérifier quelle est la performance de Routeguard lorsqu'il est utilisé contre un attaquant plus habile et de vérifier si Routeguard améliore bel et bien la

performance de 40% dans ce cas comme le prétendent les auteurs.

2.5.5 Propagation de la réputation

Lorsqu'un nœud en rencontre un second qui est inconnu, il lui assigne généralement une réputation neutre. Au fur et à la mesure qu'il interagit avec lui, il modifiera la réputation en conséquence du comportement observé. Toutefois, dans un réseau où les nœuds ont une grande mobilité, le temps de convergence de la réputation est trop grand. Utiliser la réputation que les autres nœuds donne d'un tiers peut donc être avantageux. Cette réputation risque toutefois d'être faussée. En effet, les nœuds auront souvent un avantage à mentir quant à la réputation d'un nœud. Cette sous-section présente les différentes catégories d'attaques qui peuvent survenir lors de l'échange des réputations : l'inactivité, la diffamation et la collusion (Liu et Issarny, 2004). Ensuite, différents mécanismes calculant la réputation sont introduits. Ces mécanismes permettent de mieux calculer la réputation en évaluant l'honnêteté des différents participants.

Attaques contre la propagation de la réputation

Liu et Issarny notent que les systèmes utilisant la propagation de la réputation sont souvent confrontés avec trois types de comportement de la part des usagers. Ces comportements peuvent altérer la réputation des nœuds d'une façon qui ne reflète pas leur conformité avec le protocole de routage. Voici ces trois comportements, associés avec leurs causes respectives.

Inactivité. L'inactivité survient lorsqu'un nœud décide de ne pas partager l'information qu'il possède aux autres nœuds. Propager la réputation n'apporte à priori aucun avantage à un nœud et requiert de la consommation d'énergie.

Diffamation. La diffamation survient lorsqu'un nœud propage sciemment une réputation anormalement basse à propos d'un autre nœud. Une attaque par déni de service pourrait viser des nœuds précis et les exclure d'un RAHM en abaissant suffisament leur réputation.

Collusion. Lors d'une collusion, un nœud propage une réputation anormalement élevée à propos d'un autre nœud. Cela peut être aussi bien dans le but d'aider un tiers attaquant que pour éviter la vengeance. En effet, il pourrait arriver que

les nœuds malicieux se vengent des autres lorsque leur mauvais comportement est révélé. Il deviendrait donc risqué de propager la véritable réputation de ces nœuds et serait plus avantageux de plaider en leur faveur.

Calcul de réputation et filtrage collaboratif

Le calcul de la réputation doit être réalisé de façon à ce qu'il résiste aux trois types d'attaques décrites précédemment. Ce problème est aussi rencontré dans le domaine du commerce en ligne. L'étude de ces techniques est en elle même un sujet de recherche très actif. Nous présentions ici le modèle proposé par Liu et Issarny. Seules les grandes caractéristiques et les principes sont présentés.

Les auteurs commencent par établir les différentes propriétés que devraient comprendre un système de réputations. Ces propriétés sont les suivantes :

Validité. Le système doit être efficace dans le sens où il est possible de discerner les nœuds honnêtes des nœuds malhonnêtes.

Opération distribuée. Le système ne doit pas assumer qu'une entité centrale ou parfaitement digne de confiance est présente.

Robustesse. Le système doit être résistant aux attaques décrites à la section ci-haut.

Dynamisme. Le système doit être capable de refléter la réputation des différents nœuds de façon ponctuelle.

Économique. Le système doit prendre en compte de la rareté des différentes ressources telles que la mémoire et l'énergie de chaque nœud.

Les auteurs ont donc par la suite développé leur système de réputation. Ce modèle comporte différentes caractéristiques qui le distinguent des autres systèmes de réputations. Premièrement, dans ce système, chaque nœud comporte deux réputations. La première (SRep) concerne sa capacité à donner un service de qualité, c'est-à-dire à acheminer correctement les paquets. La deuxième (RRep) est quant à elle spécifique à la capacité des nœuds de rapporter avec précision la réputation des autres nœuds. Les nœuds dérivent la première réputation SRep à partir de leur propre expérience (SExp) et des recommandations venant des autres nœuds (Rec). Les nœuds n'échangent les réputations qu'avec les nœuds ayant un bon RRep.

Ces réputations varient dans le temps. Ainsi, si un nœud a une bonne réputation à un certain temps et qu'il cesse de participer au réseau, sa réputation diminuera

tranquillement. De même, un nœud ayant une mauvaise réputation pourra faire augmenter celle-ci en agissant de façon honnête.

2.6 Synthèse des problèmes ouverts

Plusieurs CMUR ont été proposés. Lors de l'analyse de performance de ces protocoles, une amélioration significative de la performance du réseau a été notée. Il est toutefois pertinent de se questionner quant à la précision de ces analyses. En effet, l'attaquant qui est simulé est très facilement détectable. Il ne tente pas de déjouer le système utilisant la réputation. Aucun protocole n'a été testé en présence d'une ligue d'attaquants. Il est raisonnable de croire que la performance de la plupart des protocoles présentés jusqu'à date serait très inférieure en présence d'une ligue d'attaquants tentant de rester dans l'anonymat. De plus, l'influence des collisions sur la quantité de faux positifs n'est pas très exhaustive.

Ensuite, les paramètres de simulation et les niveaux de facteurs ne sont pas toujours adéquats. De plus, il est souvent difficile de vérifier l'implantation du protocole dans le simulateur. Plusieurs simulateurs sont utilisés et prouver qu'une analyse de performance est adéquate ou non est donc très difficile.

Un problème très important est causé par le fait qu'il n'existe pas à notre connaissance de modèle analytique pouvant servir à l'analyse de performance des CMUR. Un tel modèle pourrait valider l'analyse effectuée à l'aide des simulateurs. En effet, il est souvent plus simple de prouver qu'une modélisation mathématique est fausse que de trouver une erreur de programmation.

Pour combler ce manque, il serait nécessaire de définir et de formaliser mathématiquement convenablement trois aspects : l'environnement du RAHM, les indices de performance et l'attaquant. Chacun de ces aspects seront décrits dans cette section.

2.6.1 Environnement des RAHM

L'impact de l'environnement, plus particulièrement des collisions, sur les faux positifs est très important (Marti *et al.*, 2000). Toutefois, les analyses de performances ne considèrent pas adéquatement cet impact. Il faudrait donc modéliser cet impact afin de mieux analyser la performance des CMUR.

2.6.2 Indices de performance

La sélection des indices de performance peut changer radicalement les conclusions d'une analyse de performance. Ainsi, pour la très grande majorité des protocoles, les valeurs moyennes sont utilisées. Procéder de cette façon peut toutefois amener à de fausses conclusions. Considérons le scénario où les nœuds émettent tous la même quantité de trafic. Si une ligue d'attaquants veut effectuer un déni de service en rendant nul le débit de 10% des nœuds, la valeur moyenne du débit sera affectée de seulement 10%, valeur tout à fait acceptable pour la plupart des protocoles utilisant la réputation. En apparence, l'attaque est un échec même si elle a bel et bien réussi. Modéliser adéquatement les indices de performance permettrait donc de valider adéquatement le fonctionnement des CMUR.

2.6.3 Attaquants

La dernière lacune dans les analyses de performance se situe au niveau de l'attaquant et de ses actions possibles. Ainsi, les ligues d'attaquants, lorsqu'elles sont considérées, ne sont en fait qu'une duplication des nœuds malicieux. Aucune stratégie de groupe n'est implémentée. Ceux-ci ne tentent pas de passer inaperçus et sont très rapidement identifiés. Il est donc nécessaire de modéliser les attaquants ainsi que leurs différentes actions possibles.

CHAPITRE 3

Modèle de relation entre les RAHM et les attaquants

Une des principales faiblesses des articles portant sur la sécurité des réseaux ad hoc mobiles est attribuable à la mauvaise description de l'attaquant et de son interaction avec son milieu. Le comportement de l'attaquant est facilement prévisible et ne reflète en rien le comportement d'un ennemi réel. De plus, les ligues d'attaquants sont souvent absentes des tests. L'environnement qui est représenté est souvent simplifié afin de faciliter sa modélisation et l'analyse de performance. Quant aux critères de performance, ils sont souvent mal choisis. Généralement, les valeurs moyennes de ces différents critères sont utilisées. En procédant de cette façon, les auteurs ne peuvent évaluer les attaques qui ne viseraient qu'un nombre restreint de nœuds. En résumé, il est probable que la performance des différentes CMUR soit bien moindre que ce que leurs auteurs prétendent.

Toutefois, il n'existe pas de modèle analytique pour le prouver. Il est difficile de prouver si une erreur de programmation a été comise sans consulter tout le code. Par conséquent, valider l'analyse de performance peut être très difficle.

Ce chapitre propose un modèle décrivant de façon exhaustive une ligue d'attaquants agissant dans un RAHM, le contexte environnemental dans lequel cette ligue opère et décrit les critères de performance qui sont pertinents à l'analyse de performance d'une CMUR. Ce modèle analytique utilise les processus stochastiques afin de décrire l'évolution de la réputation à travers le temps.

Ce chapitre est divisé comme suit. Tout d'abord, les hypothèses utilisées pour construire le modèle sont présentées, Ensuite, le routage et la demande sont décrites formellement. Les pertes de paquets sont définies, tout comme les mécanismes de détections. Les attaques contre le routage sont définies par la suite. Ensuite, la réputation est définie. Le routage en présence d'une CMUR est défini de façon formelle, tout comme les différents indices de performance. Finalement, les caractéristi-

ques du modèle sont discutées.

3.1 Hypothèses

Différentes hypothèses sont nécessaires afin de modéliser l'environnement. Tout d'abord, les hypothèses relatives à l'environnement et à la demande sont présentées. Ensuite, les hypothèses relatives à l'attaquant sont introduites.

3.1.1 Hypothèses relatives à l'environnement

Ces hypothèses servent à définir quel est le protocole utilisé au niveau du routage, à déterminer quelles sont les CMUR qui peuvent être utilisées.

Mobilité. La connectivité entre les nœuds peut être variable. Les nœuds peuvent donc bouger et des interférences peuvent survenir. On considère toutefois que pendant un intervalle de temps correspondant à l'envoi d'un paquet de la source à la destination, la connectivité entre les nœuds est fixe.

Utilisation du protocole 802.11. Le protocole le plus utilisé dans la littérature au niveau de la couche physique et liaison est le IEEE 802.11. Cette hypothèse permet de prévoir quelles seront les pertes de paquets dues aux collisions et de mieux décrire certaines interactions entre les différents nœuds. Aussi, ce protocole exige que les liens soient bidirectionnels ce qui est également requis par la plupart des CMUR.

Non-répudiabilité et intégrité des paquets. Il est nécessaire de supposer que des mécanismes sont présents afin d'assurer ces deux propriétés. La provenance des paquets doit être connue avec certitude afin de permettre l'utilisation de la réputation. Il est pris pour acquis que la distribution des clefs relatives à l'authentification est assurée grâce à un mécanisme quelconque et qu'aucune attaque n'est possible lors de la distribution. Toutes les attaques visant l'intégrité des paquets peuvent donc être détectées.

3.1.2 Hypothèses relatives à la demande

Ces hypothèses servent à définir la nature de la demande. Ces hypothèses concernent la distribution décrivant la demande ainsi que sa variation dans le temps.

Demande sans mémoire et exponentielle. La demande, c'est-à-dire le nombre de paquets que la couche transport, via le protocole de routage, tente de transmettre suit un processus stochastique sans mémoire. C'est donc dire que si un paquet est perdu au niveau de la couche connection ou physique, il ne sera pas retransmis. Des applications pouvant être assimilées à ce comportement utilisent le protocole de transport UDP ou sont des messages ICMP. De plus, les applications ne doivent pas tenter de réémettre les informations. Des exemples de telles applications pourraient être des senseurs donnant la température à intervalle réguliers, des systèmes GPS, ou des services de nouvelles en continu, qui envoient les données sans se soucier de leur arrivée à destination.

Cette hypothèse simplifie grandement le calcul des flots entre chaque nœud et peut être étendue aux applications utilisant le protocole TCP si le taux de congestion est très faible. Toutefois, si la congestion est très élevée cette hypothèse ne pourra pas simuler adéquatement ces applications.

Par conséquent, le temps entre deux paquets consécutifs suit une loi exponentielle. Par le fait même, le nombre de paquets reçus en un intervalle de temps suit quant à lui une loi de Poisson.

Demande constante pour un intervalle de temps. La demande varie en fonction du temps. Toutefois, celle-ci varie lentement. La variation de la demande par unité de temps, donc la dérivée est négligeable par rapport à la demande.

3.1.3 Hypothèse relatives à l'attaquant

Voici les différentes hypothèses permettant la modélisation des attaquants et de leurs actions. Elles décrivent les capacités de ceux-ci ainsi que leur intention.

Attaquant interne. L'attaquant considéré est un attaquant interne, c'est-à-dire un attaquant qui a le matériel cryptographique nécessaire afin de s'authentifier.

Altération possible des adresses. Les attaquants peuvent modifier le matériel utilisé et peuvent donc changer d'adresse MAC et d'adresse IP sans toutefois pouvoir personnifier un autre nœud, étant donné la présence de mesures d'authentification.

Énergie suffisante. Les attaquants ont des ressources énergétiques suffisantes afin de mener leurs attaques à bien. Soit qu'ils ont des piles en très grande quantité ou encore qu'ils ont accès à une autre source d'énergie.

Attaque couche réseau. Seule la couche réseau est attaquée. Les attaques actives ne sont pas considérées. Les attaquants altérant des paquets peuvent être détectés de façon déterministe car la propriété de non-répudiabilité est assurée par hypothèse. Cette certitude permet de les exclure du réseau aussitôt que l'action malicieuse est détectée. Les deux attaques considérées sont donc l'action de laisser tomber les paquets délibérément et la propagation d'une réputation fausse.

- Les nœuds malicieux peuvent laisser tomber les paquets des autres nœuds. Chaque nœud possède un processus stochastique décrivant l'action de laisser tomber un paquet à un instant donné. La probabilité qu'un nœud honnête jette délibérément un paquet est nulle. Il est bon de noter que l'attaque ne se produit que contre l'acheminement des paquets. Les attaquants participent donc à la création des routes mais jettent une partie des paquets qu'ils devraient normalement acheminer.
- Les attaquants peuvent mentir au sujet de la réputation.

Matériel altérable. Par hypothèse, le matériel peut être altéré. Des solutions telles que les *Nuglets* sont donc ignorées.

3.2 Modélisation du routage et de la demande

Afin de modéliser correctement le routage et la demande des différents nœuds, trois ensembles sont définis. Le premier, $C(t)$, décrit quelle est la connectivité des différents nœuds, c'est-à-dire quels sont les nœuds qui sont à portée d'émission les uns des autres. Ensuite, $\Gamma(t)$ décrit la demande entre les nœuds. Finalement, $\gamma(t)$ correspond aux flots réels après que le protocole de routage ait été appliqué.

3.2.1 Connectivité

La connectivité détermine quels sont les nœuds qui sont à portée radio au temp t. Elle est déterminée par l'ensemble de variables aléatoires C. Les éléments de cet ensemble peuvent prendre les valeurs 0 ou 1. $C_{ij}(t) = 1$ lorsque le nœud i est capable de rejoindre le nœud j et $C_{ij}(t) = 0$ sinon. Deux nœuds sont connectés si la puissance du signal reçu par les nœuds dépasse un certain seuil. Ceci est décrit mathématiquement

de la façon suivante :

$$C_{ij} = \begin{cases} 1 & \text{si} \quad PSR_{ij} > SP \quad \text{et} \quad PSR_{ji} > SP \\ 0 & \text{Sinon} \end{cases} \tag{3.1}$$

où SP correspond à la puissance minimale pour décoder le signal et PSR_{ij} correspond à la puissance du signal émis par le nœud i et reçu par le nœud j. Le modèle de propagation des ondes qui est utilisé afin de calculer l'atténuation et la puissance du signal reçu par les nœuds est le Log Distance Path Loss Model (LDPLM)(Faria, 2006; Stuedi *et al.*, 2005). Celui-ci prend en compte la puissance du signal à un mètre de l'émetteur (PE) et une atténuation qui est linéaire en fonction du logarithme de la distance (L). Les positions servant à calculer ces distances sont des paramètres de simulations. La variable aléatoire VAA détermine quelle est l'atténuation causée par l'environnement dépendamment du milieu. Sa fonction de densité de probabilité est une loi normale de moyenne nulle et de variance égale à $STDA^2$. Le facteur d'atténuation FA est quant à lui un paramètre relié à l'environnement. Voici la fonction calculant la puissance du signal émis par i et reçu par j

$$PSR_{ij} = PE_{ij} - 10FA\log(L_{ij}) + VAA \tag{3.2}$$

La distance maximale à laquelle deux nœuds peuvent communiquer est celle où la puissance du signal reçu est SP. Alors,

$$SP = PE_{ij} - 10FA\log(L_{ij}^{\max}) + VAA$$

Si on met en évidence la distance dans l'équation précédente, il est possible de trouver la distance maximale à laquelle deux nœuds peuvent communiquer.

$$L_{ij}^{\text{Max}} = e^{\frac{VAA}{10FA}} e^{\frac{PE_{ij}-SP}{10FA}} \tag{3.3}$$

On peut substituer cette équation par une autre comprenant la distance maximale moyenne et une variable aléatoire décrivant la variation de la distance autour de cette moyenne

$$L_{ij}^{\max} = e^{VAA'} L' \tag{3.4}$$

où VAA' et L' sont des paramètres de simulation. Par conséquent, si les nœuds sont à l'intérieur de cette distance, ils pourront communiquer et leur connectivité sera égale à un.

$$C_{ij} = \begin{cases} 1 & \text{si} \quad L_{ij} < L^{\text{Max}} \\ 0 & \text{Sinon} \end{cases} \tag{3.5}$$

Ce modèle ne prend pas en compte les interférences causées par les nœuds. En effet, les nœuds qui sont à porté les uns des autres ne causeront pas d'interférence étant donné que le protocole 802.11 prévoit un mécanisme de réservation du canal. Les nœuds qui sont très éloignés pourraient toutefois causer des interférences. Toutefois, si la puissance du signal émis par ceux-ci et reçu par les nœuds i et j est beaucoup plus faible que VAA, on peut négliger leur influence dans le calcul de la connectivité. Le présent modèle néglige donc cette influence.

3.2.2 Graphe représentant le réseau

Le graphe $G(t)$ décrivant le réseau à l'instant t est constitué de l'ensemble de nœuds N, des liens représentés par la matrice de connectivité $C(t-1)$ et du poids des arcs $W(t)$. Le graphe prend en compte la topologie au moment précédent. Ainsi, si la topologie change, cette fonction n'en tiendra compte qu'au moment où les messages d'erreurs seront reçus.

$$G(t) = (N, C(t-1), W(t)) \tag{3.6}$$

Le poids des arcs détermine quel chemin sera favorisé par rapport aux autres. Il prend en compte la probabilité qu'un paquet soit perdu à cause des retransmissions ou du bruit, phénomène décrit à l'aide de la variable aléatoire E. Cette variable est définie au sein de la section 3.3. Cette fonction de poids prend en compte également la réputation des nœuds, définie comme étant la variable aléatoire R^{tot}, variable définie à la section 3.5. Cette réputation prend en compte toutes les observations, directes ou basées sur des accusés réception actifs. Elle prend aussi compte de l'avis des autres nœuds, si le protocole le permet.

Lors du calcul du poids des arcs, les pertes causées par les attaquants ne sont pas prises en compte. En effet, par hypothèse, les attaquants participent à la création des routes et donc les seules pertes de paquets sont dues aux collisions. De façon

générique, nous désignons par I_1 la fonction de calcul des poids. Nous avons donc l'équation suivante :

$$W(t) = I_1(R^{\text{tot}}{}_{ij}(t-1), C(t-1), t, E_{ij}(t-1)) \tag{3.7}$$

où $E_{ij}(t-1)$ correspond à la variable aléatoire décrivant si un paquet allant du nœud i vers le nœud j est jeté par le nœud j au temps $t-1$. La façon de calculer ces poids est propre à chaque protocole de routage et CMUR. Il est bon de noter que cette fonction tient compte du parcours. En effet, certaines de ces CMUR donnent un poids différent en fonction du chemin et évitent ainsi qu'une route comprenne deux nœuds malicieux un à la suite de l'autre.

Lorsqu'aucune CMUR n'est présente, seul l'environnement modifiera le poids des arcs, la réputation n'entrant pas en ligne de compte. Sinon, la réputation des nœuds ainsi que la topologie sera prise en compte dans le calcul du poid.

Par exemple, pour le protocole DSR, le poids des arcs sera égal à 1 si le lien est utilisable ou à une valeur très grande si le lien ne l'est pas. Ainsi, la recherche du plus court chemin dans le graphe tentera de prendre un trajet qui ne contient pas de liens surchargés, quitte à parcourir une plus grande distance dans le réseau. Si aucun lien n'est surchargé, le trajet choisi sera celui comprenant le moins de sauts. Donc, dans le cas de DSR, le poids des arcs est calculé à l'aide de la fonction suivante :

$$W(t) = 1 + E_{ij}(t-1)|N| \tag{3.8}$$

où $|N|$ est le nombre de nœuds dans le réseau.

3.2.3 Demande

La demande est représentée par un processus stochastique, plus précisément par un processus de comptage. La somme des trames que l'on a voulu transmettre par le protocole de routage entre le temps 0 et le temps t, qu'elles se soient rendues à destination ou non, est représenté par la varible aléatoire $Co(t)$.

Comme, par hypothèse, ce processus est sans mémoire, le nombre moyen de trames ayant à être transmises est égal à

$$\langle Co_{ij}(t) \rangle \ = \Gamma_{ij}(t)t \tag{3.9}$$

où $\Gamma_{ij}(t)$ varie très lentement par hypothèse.

L'ensemble de paramètres $\Gamma(t)$ détermine combien de paquets par seconde doivent être acheminés entre une source i et une destination j en moyenne au temps t.

L'ensemble de variables aléatoires $\gamma_{ij}(t)$ détermine combien de paquets sont échangés en moyenne entre un nœud i et un autre nœud j entre les instants t et $t+1$ en tenant compte de $\Gamma_{ij}(t)$, de la topologie, du protocole de routage et des pertes de paquets au temps t, représentées par l'ensemble de variables aléatoires T.

La matrice $\gamma_{ij}(t)$ est donc obtenue en trouvant les plus courts chemins entre les nœuds i et j dans le graphe G à l'aide de la fonction c que voici :

$$\gamma_{ij}(t) = c(G(t), \Gamma(t), T(t)) \tag{3.10}$$

Le flot entre les nœuds est par la suite calculé à l'aide de ces routes. Cette fonction prend en compte tous les paquets qui seront jetés, peu importe la raison. Dans le cas où le protocole DSR est utilisé, l'algorithme de la fonction c se calquera sur celui de DSR où les routes sont celles où le nombre de sauts est le moins élevé.

3.3 Pertes de paquets

Cette section décrit comment est modélisée la perte des paquets ainsi que la cause de chaque perte. Il est nécessaire de prévoir quelle est la quantité de paquets qui sera perdu par chaque type de nœud, qu'il soit attaquant ou honnête. Les pertes de paquets attribuables à l'environnement, donc aux collisions, sont donc modélisées, tout comme celles dues aux attaquants et à la réputation.

3.3.1 Pertes de paquets dues à la réputation

Certaines CMUR fixent une réputation minimale à maintenir pour les nœuds afin que leurs paquets soient retransmis par les autres nœuds. Une partie des paquets qui sont reçus par les différents nœuds seront donc jetés à cause de la trop faible réputation d'un des nœuds sur la route. Ce rejet dépend d'une fonction, notée I_2, propre à chaque CMUR utilisée. Par exemple, Routeguard exige que tous les nœuds qui font partie de la route ne soient pas malicieux. Si un nœud est malicieux, le paquet est jeté. Aussi, si la source du paquet n'est pas membre du réseau, le paquet sera également jeté.

Tout d'abord, la variable aléatoire $F_{i,j,H}(t)$ détermine si le nœud i jette un paquet transitant par la route H au temps t alors qu'il aurait dû autrement acheminer le paquet vers le nœud j. Si $F_{i,j,H}(t) = 1$, le paquet a été jeté, sinon, $F_{i,j,H}(t) = 0$. La probabilité que cette perte de paquet survienne suit une fonction qui est propre à chaque CMUR et qui dépend de la réputation des nœuds sur la route selon le jugement de i. La valeur ${R^{\text{tot}}}_{ij}(t)$ qui représente la réputation de j telle que perçue par i au temps t est décrite en détail à la section 3.5.

Une route quelconque où les nœuds i et j sont consécutifs et où n autres nœuds sont présents est notée de la façon suivante :

$$H = \{k_1, k_2, \ldots, i, j, \ldots, k_n\} \tag{3.11}$$

Par conséquent, la probabilité de perdre un paquet sur une telle route est égale à

$$\Pr(F_{i,j,H}(t) = 1) = I_2(t, {R^{\text{tot}}}_{ik_1}(t), {R^{\text{tot}}}_{ik_2}(t), \ldots, {R^{\text{tot}}}_{ij}(t), \ldots, {R^{\text{tot}}}_{ik_n}(t)) \tag{3.12}$$

Le nœud i n'est pas nécessairement l'origine du paquet et le nœud j n'est pas nécessairement sa destination.

La variable aléatoire $F_{ij}(t)$ représente l'événement où le nœud i jette un paquet devant être acheminé vers le nœud j à cause de la réputation insuffisante d'un nœud présent sur la route empruntée par ce paquet. Si $F_{ij}(t) = 1$ si le paquet a été jeté, sinon $F_{ij}(t) = 0$.

La probabilité qu'un paquet allant du nœud i au nœud j soit rejeté parce que la réputation d'un nœud sur la route est trop faible est égale à la proportion des paquets qui passent par une route H où $F_{i,j,H}(t) = 1$

Si on note NbH le nombre de routes où les nœuds i et j sont consécutifs, H_n comme étant la $n^{\text{ième}}$ route où ces nœuds sont consécutifs, $\gamma_{H_n}(t)$ comme représentant le flot total passant par la route H_n, cette probabilité peut s'exprimer ainsi :

$$\Pr(F_{ij}(t) = 1) = \frac{\sum_{n=1}^{NbH} F_{i,j,H_n}(t)\gamma_{H_n}(t)}{\gamma_{ij}(t)} \tag{3.13}$$

3.3.2 Attaque contre l'acheminement des paquets

L'intention des nœuds est déterminé par l'ensemble de variables aléatoires $J(t)$ qui, en fonction du temps, décrit si un nœud jete ou non une trame qui est à destina-

tion d'un autre nœud. Les valeurs que peuvent prendre ces variables aléatoires sont constitués de l'ensemble $\{0, 1\}$.

Si $J_{ij}(t) = 0$, le nœud i a l'intention d'acheminer le paquet vers le nœud j tandis que si $J_{ij}(t) = 1$, le nœud i a l'intention de jeter le paquet vers le nœud j. Par conséquent, si le nœud i n'est pas malicieux,

$$\Pr(J_{ij}(t) = 0) = 1, \ \forall j \in N, \ \forall t \geq 0 \tag{3.14}$$

3.3.3 Pertes de paquets dues à l'environnement

Les réseaux sans-fils sont très susceptibles de connaître de forts taux de collisions. Ainsi, une quantité non négligeable de paquets sera jetée involontairement par les nœuds. La réputation des nœuds honnêtes pourra donc être très basse si le réseau est congestionné. Afin de modéliser la réputation, il est donc primordial de modéliser ces pertes de paquets. Les collisions peuvent avoir lieu si un nœud adjacent à i ou j reçoit ou envoie une trame quelconque au même instant où le nœud i tente de réserver le canal et qu'une situation semblable se répète. Le nœud i pourrait alors dépasser le nombre maximal de retransmissions pour envoyer le paquet et le jeter. Une autre cause est que les nœuds i et j ne puissent pas communiquer au temps t, c'est-à-dire que $C_{ij}(t) = 0$.

Soit $E_{ij}(t)$ la variable aléatoire décrivant la perte de paquets due aux collisions. $E_{ij}(t) = 0$ si le paquet allant du nœud i au nœud j n'a pas été perdu au temps t et $E_{ij}(t) = 1$ si le paquet a été perdu.

La probabilité $\Pr(E_{ij}(t) = 1)$ dépend tout d'abord de la connectivité au temps t. En effet, si lors de la création des routes, les nœuds i et j étaient à proximité et que par après, ils se sont éloignés ou si leur ratio signal sur bruit est devenu trop faible, le paquet ne sera pas acheminé. Par conséquent,

$$\Pr(E_{ij}(t) = 1 | C_{ij}(t) = 0) = 1 \tag{3.15}$$

Cependant, il n'existe pas de modèle simple et complet pouvant calculer la probabilité $\Pr(E_{ij}(t) = 1 | C_{ij}(t) = 1)$. Certains modèles prennet en compte que tous les flots entre les nœuds sont exactement les mêmes (Özdemir et McDonald, 2005). Comme la modélisation du mécanisme de réservation du canal dans le protocole 802.11 est un sujet de recherche en lui-même dépassant largement le cadre de ce mémoire, une

version allégée du protocole 802.11 est utilisée. Ainsi, le protocole 802.11 sera utilisé sans les mécanismes de réservation du canal et avec une limite de retransmission égale à 0, ce qui veut dire que si le message ne peut pas être transmis du premier coup, il sera jeté. En utilisant un modèle plus complet, il serait possible de modéliser le 802.11 ainsi que tous ses mécanismes. On prend donc trois hypothèses.

1. Il est très rare que deux nœuds commencent à émettre un paquet simultanément.
2. La quantité d'interférences causées par les données est beaucoup plus grande que la quantité de trafic provenant des mécanismes de gestion des routes. Afin de prendre en compte ces mécanismes, il serait nécessaire de savoir exactement quel mécanisme de création des routes est utilisé.
3. Le flot sortant d'un nœud est toujours inférieur à la capacité du canal. En effet, si la capacité du canal est dépassée, des paquets pourraient rester trop longtemps dans la file d'émission et être jetés pour cette raison. Comme chaque protocole a son propre temps limite et que les scénarios utilisés lors de l'analyse de performance du modèle ont des flots très inférieurs à la capacité du canal, on utilise cette hypothése.

Dans le protocole 802.11, pour émettre un paquet, un nœud i doit écouter pendant une fenêtre de temps notée DIFS et de durée T_{DIFS}. Si aucun nœud n'émet pendant ce temps, il pourra émettre. Le temps moyen d'émission de la trame est représenté par la variable T_{DATA}. Si le nœud destination j reçoit correctement la trame, ce qui signifie qu'aucun autre nœud dans sa portée n'a émis de trame en même temps, il émettra un accusé réception vers le nœud émetteur. Comme aucune retransmission n'est modélisée, si une trame est émise pendant la fenêtre de temps T_{DIFS} ou pendant l'émission du paquet, il y aura interférence. Il est nécessaire de noter qu'on prend en compte qu'aucun paquet ne peut être émis pendant le temps nécessaire à l'émission de l'accusé réception, temps représenté par la constante T_{ACK}. En effet, comme le temps d'émission de cet accusé est plus petit que le temps T_{DIFS}, il est impossible qu'un nœud adjacent au nœud i émette, étant donné que s'il tente d'émettre en même temps que i, il y aura collision, et il ne pourra émettre qu'après une période de temps T_{DIFS}. De la même façon, si un nœud à portée de j émet pendant que i émet, il y aura interférence et s'il émet après l'émission de i, il devra attendre une période de temps T_{DIFS}.

Une collision peut donc survenir à deux moments, c'est-à-dire lorsque le nœud i

écoute pour savoir si le médium est occupé et lorsque le nœud i émet le paquet. Deux types de trames peuvent causer cette interférence. Un paquet contenant des données peut la causer tout comme une trame 802.11 d'accusé réception.

Il y a donc quatre événements qui sont présents, chacun d'entre eux étant modélisés par une variable aléatoire. Celles-ci modélisent

a. La collision lors du DIFS causée par un paquet de données
b. La collision lors du DIFS causée par un accusé réception.
c. La collision lors de l'émission des données causée par l'émission des données.
d. La collision lors de l'émission des données causée par l'émission d'un accusé réception.

Voici la description de ces variables aléatoires, décrites par S^{a}, S^{b}, S^{c}, S^{d}, ainsi que le calcul des probabilités reliées à chacune d'entre elles.

Collision pendant le DIFS par un paquet de données

La variable aléatoire S^{a} décrit si une collision causée par un paquet de données a lieu lors du DIFS. Si $S^{\mathrm{a}} = 1$, une collision a lieu, sinon $S^{\mathrm{a}} = 0$.

Le taux de paquets de données, émis par les autres nœuds, donné par ${\Lambda^{\mathrm{a}}}_i(t)$ est égal au taux de paquets devant être acheminés par les nœuds voisins k de i vers les nœuds x. Ces paquets ne doivent pas être jetés pour une des raisons suivantes :

- Le nœud k a détecté une collision lors du DIFS, défini par les événements ${S^{\mathrm{a}}}_k(t) = 1$ et ${\Lambda^{\mathrm{b}}}_k(t) = 1$, lors de la transmission du paquet vers le nœud x.
- Le nœud k a jeté le paquet délibérément, représenté par l'événement $J_{kx}(t) = 1$.
- La réputation d'un nœud sur la route est trop basse, ce qui est représenté par l'événement $F_{kx}(t) = 1$.

Il est bon de noter que les paquets que le nœud k tente de transmettre mais qui ne sont pas reçus par le nœud x sont comptés dans cette somme qui peut être exprimé mathématiquement de la façon suivante :

$$ {\Lambda^{\mathrm{a}}}_i(t) = \sum_{k|k\neq i, C_{ki}(t)=1} \sum_{x} (1 - {S^{\mathrm{a}}}_{kx}(t))(1 - {S^{\mathrm{b}}}_{kx}(t))(1 - J_{kx}(t))(1 - F_{kx}(t))\gamma_{kx}(t) $$

Le temps pendant lequel il ne doit pas y avoir de tentative d'émission est égal à $T_{\mathrm{DIFS}} + T_{\mathrm{DATA}}$. En effet, si un nœud commence à émettre un paquet moins de T_{DATA}

avant que le nœud i commence sa période d'écoute, il y aura collision, le nœud n'ayant pas terminé d'émettre quand i commencera l'écoute. Ensuite, si le nœud commence à émettre pendant la période de temps T_{DIFS}, il y aura également collision.

Comme la distribution du temps entre deux paquets suit une loi exponentielle par hypothèse, le nombre moyen de paquets émis par intervalle de temps suit quant à lui une loi de Poisson. La probabilité que ce nombre de paquets, représenté par la variable nb soit plus petit ou égal à un nombre de paquets nb' pendant le temps $T_{\text{DIFS}} + T_{\text{DATA}}$ est donc égal à

$$\Pr(nb \leq nb') = \sum_{n=0}^{nb'} \frac{(\Lambda^{\text{a}}{}_i(t)(T_{\text{DIFS}} + T_{\text{DATA}})^n}{n!} e^{-\Lambda^{\text{a}}(T_{\text{DIFS}}+T_{\text{DATA}})} \tag{3.16}$$

Par conséquent, la probabilité qu'aucun paquet ne soit émis, c'est-à-dire que $nb = 0$ est égale à

$$\Pr(S^{\text{a}}{}_i(t) = 0) = \Pr(nb = 0) = e^{-\Lambda^{\text{a}}{}_i(t)(T_{\text{DATA}}+T_{\text{DIFS}})} \tag{3.17}$$

Collision pendant le DIFS par un accusé réception 802.11

La variable aléatoire S^{b} décrit si une collision causée par un accusé de réception a lieu lors du DIFS. Si $S^{\text{b}} = 1$, une collision a lieu, sinon $S^{\text{b}} = 0$.

Le taux de paquets de données émis par les autres nœuds k est égal au taux de paquets devant être acheminés vers les nœuds voisins de i qui ne sont pas jetés pour une des raisons suivantes :

- Le nœud x n'a pas réussi à transmettre le paquet au nœud k à cause des interférences, représenté par l'événement $E_{xk}(t) = 1$.
- Le nœud x a jeté le paquet délibérément, représenté par l'événement $J_{xk}(t) = 1$.
- La réputation d'un nœud sur la route est trop basse, c'est-à-dire que $F_{xk}(t) = 1$

Ceci est exprimé mathématiquement de la façon suivante :

$$\Lambda^{\text{b}}{}_i(t) = \sum_{k|C_{ki}(t)=1, k\neq i} \sum_x (1 - E_{xk}(t))(1 - J_{xk}(t))(1 - F_{xk}(t))\gamma_{xk}(t)$$

Pour qu'un paquet soit compté dans cette somme, il doit avoir été reçu par le nœud k, et donc que $E_{xk} = 0$, contrairement aux collisions causées par les paquets de données. En effet, dès qu'un paquet de données est émis, il peut potentiellement causer une

collision. Par contre, pour qu'un accusé réception soit émis par le nœud k, $E_{xk} = 0$, ce qui explique la différence dans le calcul des collisions causé par les paquets de données et ceux causés par les accusés réception 802.11.

Le temps pendant lequel il ne doit pas y avoir de tentative d'émission est égal à $T_{\text{DIFS}} + T_{\text{ACK}}$ où T_{ACK} correspond au temps d'émission d'un accusé réception. En effet, si un nœud commence à émettre un accusé de réception moins de T_{ACK} avant que le nœud i commence sa période d'écoute, il y aura collision, le nœud n'ayant pas terminé d'émettre quand i commencera l'écoute. Ensuite, si le nœud commence à émettre pendant la période de temps T_{DIFS}, il y aura également collision.

$$\Pr({S^{\text{b}}}_i(t) = 0) = e^{-{\Lambda^{\text{b}}}_i(t)(T_{\text{ACK}}+T_{\text{DIFS}})} \tag{3.18}$$

Collision pendant l'émission des données par un paquet de données

La variable aléatoire S^{c} décrit si une collision causée par un paquet de données a lieu lors de l'émission des données. Si $S^{\text{c}} = 1$, une collision a lieu, sinon $S^{\text{c}} = 0$.

Le taux de paquets de données émis par les autres nœuds est égal au taux de paquets devant être acheminés par les nœuds k voisins de j mais pas de i qui ne sont pas jetés pour une des raisons suivantes.

- Le nœud k a détecté une collision lors du DIFS, défini par les événements ${S^{\text{a}}}_k(t) = 1$ et ${\Lambda^{\text{b}}}_k(t) = 1$, lors de la transmission du paquet vers le nœud x.
- Le nœud k a jeté le paquet délibérément, représenté par l'événement $J_{kx}(t) = 1$.
- La réputation d'un nœud sur la route est trop basse, ce qui est représenté par l'événement $F_{kx}(t) = 1$.

En effet, un nœud voisin de i n'émettra pas pendant que i émet étant donné qu'il commencera par attendre un temps DIFS avant d'émettre et qu'il est, par hypothèse, impossible que deux nœuds commencent à émettre exactement en même temps.

Ce taux est exprimé mathématiquement de la façon suivante :

$${\Lambda^{\text{c}}}_{ij}(t) = \sum_{k|C_{ki}(t)=0,C_{kj}(t)=1} \sum_{x} (1 - {S^{\text{a}}}_{kx}(t))(1 - {S^{\text{b}}}_{kx}(t))(1 - J_{kx}(t))(1 - F_{kx}(t))\gamma_{kx}(t)$$

Le temps pendant lequel il ne doit pas y avoir de tentative d'émission est égal à $T_{\text{DATA}} + T_{\text{DATA}}$. En effet, si un nœud commence à émettre un paquet moins de T_{DATA}

avant que le nœud i commence sa période d'écoute, il y aura collision, le nœud n'ayant pas terminé d'émettre quand i commencera l'écoute. Ensuite, si le nœud commence à émettre pendant la période de temps T_{DATA}, il y aura également collision.

$$\Pr({S^{\text{c}}}_{ij}(t) = 0) = e^{-{\Lambda^{\text{c}}}_{ij}(t)(T_{\text{DATA}}+T_{\text{DATA}})} \tag{3.19}$$

Collision pendant l'émission des données par un accusé réception 802.11

La variable aléatoire S^{d} décrit si une collision causée par un accusé de réception a lieu lors de l'émission des donnnées. Si $S^{\text{d}} = 1$, une collision a lieu, sinon $S^{\text{d}} = 0$.

Le taux de paquets de données émis par les autres nœuds est égal au taux de paquets devant être acheminés par les nœuds voisins k de j mais pas voisins de i qui ne sont pas jetés pour une des raisons suivantes :

- Le nœud x n'a pas réussi à transmettre le paquet au nœud k à cause des interférences, représenté par l'événement $E_{xk}(t) = 1$.
- Le nœud x a jeté le paquet délibérément, représenté par l'événement $J_{xk}(t) = 1$.
- La réputation d'un nœud sur la route est trop basse, c'est-à-dire que $F_{xk}(t) = 1$

Ceci est exprimé mathématiquement de la façon suivante :

$${\Lambda^{\text{d}}}_{ij}(t) = \sum_{k|C_{kj}(t)=1,C_{ki}(t)=0} \sum_{x} (1 - E_{kx}(t))(1 - J_{kx}(t))(1 - F_{kx}(t))\gamma_{xk}(t)$$

Le temps pendant lequel il ne doit pas y avoir de tentative d'émission est égal à $T_{\text{ACK}} + T_{\text{DATA}}$. En effet, si un nœud commence à émettre un paquet moins de T_{DATA} avant que le nœud i commence sa période d'écoute, il y aura collision, le nœud n'ayant pas terminé d'émettre quand i commencera l'écoute. Ensuite, si le nœud commence à émettre pendant la période de temps T_{DIFS}, il y aura également collision.

$$\Pr({S^{\text{d}}}_{ij}(t) = 0) = e^{-{\Lambda^{\text{d}}}_{ij}(t)(T_{\text{DATA}}+T_{\text{ACK}})} \tag{3.20}$$

Probabilité de pertes dues à l'environnement

Les quatres probabilités calculées précédemment permettent de déterminer la probabilité qu'un paquet ne soit pas jeté pour des raisons environnementales. Si aucun paquet n'est émis pendant le DIFS par un nœud adjacent à i ni pendant l'émission

des données par un nœud adjacent à j, il n'y aura pas de pertes de paquets. Donc, la probabilité que le paquet ne soit pas bloqué par l'environnement est égale à :

$$\Pr(E_{ij} = 0 | C_{ij} = 1) = \Pr([S^{a}{}_{i} = 0], [S^{b}{}_{i} = 0], [S^{c}{}_{ij} = 0], [S^{d}{}_{ij} = 0]) \tag{3.21}$$

En substituant les variables S^{a}, S^{b}, S^{c}, S^{d} par leur valeur calculées dans les équations 3.17 à 3.20, on obtient l'équation suivante :

$$\Pr(E_{ij} = 0 | C_{ij} = 1) = e^{-(T_{\text{DATA}}(\Lambda^{a}{}_{i} + 2\Lambda^{c}{}_{ij}(t) + \Lambda^{d}{}_{ij}) + T_{\text{ACK}}(\Lambda^{b}{}_{i} + \Lambda^{d}{}_{ij}(t)) + T_{\text{DIFS}}(\Lambda^{a}{}_{i} + \Lambda^{b}{}_{i}))} \tag{3.22}$$

Par conséquent, la probabilité de pertes dues à l'environnement est égale à :

$$\Pr(E_{ij} = 1 | C_{ij} = 1) = 1 - \Pr([S^{a}{}_{i} = 0], [S^{b}{}_{i} = 0], [S^{c}{}_{ij} = 0], [S^{d}{}_{ij} = 0]) \tag{3.23}$$

De la même façon, en substituant les variables S^{a}, S^{b}, S^{c}, S^{d} par leur valeur calculées dans les équations 3.17 à 3.20, on obtient l'équation suivante :

$$\Pr(E_{ij} = 1 | C_{ij} = 1) = 1 - e^{-(T_{\text{DATA}}(\Lambda^{a}{}_{i} + 2\Lambda^{c}{}_{ij}(t) + \Lambda^{d}{}_{ij}) + T_{\text{ACK}}(\Lambda^{b}{}_{i} + \Lambda^{d}{}_{ij}(t)) + T_{\text{DIFS}}(\Lambda^{a}{}_{i} + \Lambda^{b}{}_{i}))} \tag{3.24}$$

3.3.4 Probabilité de laisser tomber un paquet

L'événement $T_{ij}(t)$ décrit le fait que le nœud j n'achemine pas un paquet en provenance du nœud i au temps t, volontairement ou non, alors que ce paquet était destiné à être acheminé. Si le paquet est jeté, $T_{ij}(t) = 1$, sinon, $T_{ij}(t) = 0$. Les paquets envoyés directement entre une source et une destination sont aussi compris dans cet événement.

La probabilité associée à l'événement $T_{ij}(t) = 0$ est égale à la probabilité que l'un des événements suivants arrivent :

- Le rejet intentionnel d'un paquet par j, ce qui correspond à une attaque.
- La perte d'un paquet causée par une réputation trop faible d'un nœud autre que j, cette réputation étant calculée par j.
- La perte d'un paquet causée par l'environnement. Cette perte peut avoir lieu alors que le nœud i tente d'acheminer le paquet au nœud j ou lorsque le nœud

j tente de l'acheminer au nœud k, k étant le prochain nœud sur la route.

La probabilité qu'un paquet soit jeté par le nœud j dépend donc du nœud qui le suit et par conséquent, de la route qui est empruntée par le paquet. Nous reprenons donc la définition de la route décrite par l'équation 3.11, c'est-à-dire qu'une route quelconque où les nœuds i et j sont consécutifs et où n autres nœuds sont présents est notée :

$$H = \{k_1, k_2, \ldots, i, j, \ldots, k_n\}$$

Sur une route H, le flot total qui y passe est noté $\gamma_H(t)$. Pour cette route, la probabilité que j jette un paquet est égale à

$$\Pr(T_{ijH} = 1) = \Pr([J_{jk}(t) = 1] \cup [E_{jk}(t) = 1] \cup [F_{jk}(t) = 1] \cup [E_{ij}(t) = 1]) \quad (3.25)$$

où k correspond au nœud suivant le nœud j dans la route.

La probabilité qu'un paquet allant du nœud i au nœud j soit jeté est égale à la proportion des paquets qui sont jetés et qui passent par une route H où $T_{ijH}(t) = 1$, ou qui vont directement de la source à la destination. Voici donc cette probabilité décrite mathématiquement :

$$\Pr(T_{ij}(t) = 1) = \frac{\sum_n^{NbH} T_{ijH_n}\gamma_{H_n}(t)}{\gamma_{ij}(t)} + \frac{E_{ij}(t)\Gamma_{ij}(t)}{\gamma_{ij}(t)} \quad (3.26)$$

Il est bon de noter que lorsqu'un paquet va directement de la source à la destination, on considère qu'il ne peut pas être jeté parce que la réputation d'un nœud est trop basse ou parce qu'un attaquant est présent. Seul l'environnement peut être la cause de cette perte.

3.4 Détection

Cette section décrit comment la détection des paquets jetés est modélisée par le modèle. Cette détection ne dépend pas de l'attaquant ni du protocole de gestion des réputations. Par contre, la détection des paquets dépend des flots des nœuds environnants et du protocole utilisé à la couche liaison.

3.4.1 Probabilité de détecter correctement un paquet

Les collisions entre les paquets peuvent causer des faux positifs et des faux négatifs (Marti *et al.*, 2000). Dans l'exemple suivant, cinq nœuds (i, j, k, q, r) sont présents. La route empruntée par les paquets est $H = \{k_1, k_2, \ldots, i, j, \ldots, k_n\}$ où le nœud k est le nœud suivant le nœud j. Le nœud i tente donc d'envoyer un paquet à destination de k en le faisant acheminer par j. Le nœud q est un nœud quelconque causant interférence en émettant au nœud r.

L'événement $D_{ij}(t)$ représente le fait de détecter correctement l'acheminement ou le non acheminement d'un paquet au temps t. Si $D_{ij}(t) = 1$ l'action a été bien détectée, si $D_{ij}(t) = 0$, l'action a été mal détectée. Voici donc la description de ces événements. Dans les deux cas, le scénario envisagé est celui où le nœud source i écoute afin de savoir si le nœud j achemine bien le message au nœud destination k. Un nœud q cause les interférences en émettant à un nœud quelconque r.

Paquet incorrectement détecté comme jeté.

La probabilité que la détection effectuée par le nœud i envers le nœud j est mauvaise et que le paquet était acheminé est égale à

$$\Pr(D_{ij}(t) = 0 | T_{ij}(t) = 0) = \frac{\sum_{n=0}^{NbH} \Pr(D_{ijH_n}(t) = 0 | T_{ijH_n}(t) = 0)\gamma_{H_n}(t)}{\gamma_{ij}(t)} \quad (3.27)$$

où $\Pr(D_{ijH_n}(t) = 0)$ correspond à l'événement se produisant lorsque un nœud quelconque q émet à destination d'un nœud quelconque r et que ce message est reçu par le nœud i pendant que le nœud j achemine le paquet au nœud k sur une route H_n. La variable $\gamma_{H_n}(t)$ correspond au flot de i vers k et acheminé par j sur une route donnée, telle que définie à la section 3.3. La variable aléatoire $T_{ijH_n}(t)$ est elle aussi définie à la section 3.3.

Cet événement dépend donc des flots reçus ou émis par l'ensemble des nœuds voisins de i mais pas voisins de j ou k. En effet, l'émission du nœud q ne doit pas être reçue par le nœud k sinon, cela ne constituerait plus un faux positif, le message ne s'étant pas rendu. De plus, comme le protocole 802.11 est utilisé, les nœuds q et r ne doivent pas être à portée du nœud j. En effet, si le nœud j émet le paquet, les nœuds q et r le détecteront pendant la période DIFS. On présume également qu'il est très peu probable que les deux nœuds commencent l'émission exactement en même

temps.

La probabilité $\Pr(D_{ijH_n}(t) = 0 | T_{ijH_n}(t) = 0)$ est égale à la probabilité qu'il y ait au moins un paquet qui soit émis par un voisin de i pendant le temps de transmission, donc égale à 1 moins la probabilité qu'il n'y en ait aucun. Ces paquets peuvent être des paquets de données ou des accusés réception.

Le cheminement pour trouver les paquets causant les interférences, les temps pendant lesquels ils causent ces interférences et la probabilité d'interférence est le même que celui utilisé pour le calcul des pertes de paquets relatives à l'environnement.

La quantité de paquets de données causant des interférences est égale à

$$\lambda^{\text{DATA}}{}_{ijH_n}(t) = \sum_{z|(C_{zi}=1, C_{zj}=0, C_{zk}=0)} \sum_{x|\Lambda^{\text{a}}{}_{zx}(t)=0, \Lambda^{\text{b}}{}_{zx}(t)=0, J_{zx}(t)=0, F_{zx}(t)=0} \gamma_{zx}(t)$$

tandis que la quantité d'accusés réception causant des interférences est égale à

$$\lambda^{\text{ACK}}{}_{ijH_n}(t) = \sum_{z|(C_{zi}=1, C_{zj}=0, C_{zk}=0)} \sum_{x|E_{xz}(t)=0, J_{xz}(t)=0, F_{xz}(t)=0} \gamma_{xz}(t)$$

Le temps pendant lequel il ne doit pas y avoir de données est égal à $T_{\text{DATA}} + T_{\text{DATA}}$ tandis que le temps pendant lequel il ne doit pas y avoir d'accusés réception est égal à $T_{\text{DATA}} + T_{\text{ACK}}$.

Par conséquent la probabilité qu'un nœud émette pendant que j achemine le paquet est égale à

$$\Pr(D_{ijH_n}(t) = 0 | T_{ijH_n}(t) = 0) = 1 - e^{-(\lambda^{\text{DATA}}{}_{ijH_n}(T_{\text{DATA}}+T_{\text{DATA}}) + \lambda^{\text{ACK}}{}_{ijH_n}(T_{\text{DATA}}+T_{\text{ACK}}))} \tag{3.28}$$

Paquet incorrectement détecté comme acheminé

Cette situation, représentée mathématiquement par l'événement $[D_{ij}(t) = 0 | T_{ij}(t) = 1]$, se produit lorsque le nœud q émet à destination d'un nœud quelconque r et que ce message est reçu par k pendant que le nœud j envoyait son message au nœud k. Il est également nécessaire que les nœuds i et q ne soient pas à portée l'un de l'autre. Le message en question peut tout aussi bien être des données

qu'un accusé réception. La probabilité que cette situation survienne est égale à

$$\Pr(D_{ij}(t)=0|T_{ij}(t)=1)=\frac{\sum_{n}^{NbH}\Pr(D_{ijH_n}(t)=0|T_{ijH_n}(t)=1)\gamma_{ijH_n}(t)}{\gamma_{ij}(t)} \quad (3.29)$$

où $\Pr(D_{ijH_n}(t)=0|T_{ijH_n}(t)=1)$ correspond à la probabilité qu'il y ait une interférence de causée par le nœud k sur la route H_n.

Le cheminement pour trouver les paquets causant les interférences, les temps pendant lesquels ils causent ces interférences et la probabilité d'interférence est le même que celui utilisé pour le calcul des paquets détectés comme étant jetés.

La quantité de paquets de données causant des interférences est égale à

$$\bar{\lambda}^{\mathrm{DATA}}{}_{ijH_n}(t)=\sum_{z|(C_{zi}=0,C_{zj}=0,C_{zk}=1)}\ \sum_{x|\Lambda^{\mathrm{a}}{}_{zx}(t)=0,\Lambda^{\mathrm{b}}{}_{zx}(t)=0,J_{zx}(t)=0,F_{zx}(t)=0}\gamma_{zx}(t)$$

tandis que la quantité d'accusés réception causant des interférences est égale à

$$\bar{\lambda}^{\mathrm{ACK}}{}_{ijH_n}(t)=\sum_{z|(C_{zi}=0,C_{zj}=0,C_{zk}=1)}\ \sum_{x|E_{xz}(t)=0,J_{xz}(t)=0,F_{xz}(t)=0}\gamma_{xz}(t)$$

Le temps pendant lequel il ne doit pas y avoir de données est égal à $T_{\mathrm{DATA}}+T_{\mathrm{DATA}}$ tandis que le temps pendant lequel il ne doit pas y avoir d'accusés réception est égal à $T_{\mathrm{DATA}}+T_{\mathrm{ACK}}$.

Par conséquent la probabilité qu'un nœud émette pendant que j achemine le paquet est égale à

$$\Pr(D_{ijH_n}(t)=0|T_{ijH_n}(t)=1)=1-e^{-(\bar{\lambda}^{\mathrm{DATA}}{}_{ijH_n}(T_{\mathrm{DATA}}+T_{\mathrm{DATA}})+\bar{\lambda}^{\mathrm{ACK}}{}_{ijH_n}(T_{\mathrm{DATA}}+T_{\mathrm{ACK}}))} \quad (3.30)$$

3.4.2 Probabilité de détecter qu'un paquet a été jeté

La probabilité d'observer l'acheminement d'un paquet dépend de la probabilité qu'il soit jeté et de la probabilité que l'on observe correctement l'action.

Soit l'événement $TO_{ij}(t)$ décrivant l'action de détecter qu'un paquet a été jeté, que la détection soit bonne ou non. Si $TO_{ij}(t)=1$, on a détecté que le paquet a été jeté, $TO_{ij}(t)=0$ sinon.

La probabilité d'observer qu'un paquet a été jeté est égale à la probabilité qu'un

paquet ait été jeté et que l'observation ait été correcte ou qu'un paquet n'ait pas été jeté mais que l'observation ait été faussée. Voici cette probabilité décrite de façon formelle :

$$\Pr(TO_{ij}(t) = 1) = \Pr(D_{ij}(t) = 1, T_{ij}(t) = 1 \cup D_{ij}(t) = 0, T_{ij}(t) = 0) \quad (3.31)$$

où $\Pr(D_{ij}(t) = 1)$ peut être calculée à l'aide des équations 3.25, 3.26,3.27, 3.28,3.29 et 3.30. La probabilité $\Pr(T_{ij}(t) = 1$ peut être calculée à l'aide des équations 3.25 et 3.26.

Comme $\Pr((D_{ij}(t) = 1, D_{ij}(t) = 0) = 0$ on peut réécrire cette probabilité comme étant égale à :

$$\Pr(TO_{ij}(t) = 1) = \Pr(D_{ij}(t) = 1, T_{ij}(t) = 1) + \Pr(D_{ij}(t) = 0, T_{ij}(t) = 0) \quad (3.32)$$

La probabilité d'observer qu'un paquet a été jeté est quant à elle égale à la probabilité qu'un paquet ait été jeté et que l'observation soit exacte ou que le paquet n'ait pas été jeté et que l'observation soit inexacte.

$$\Pr(TO_{ij}(t) = 0) = \Pr(D_{ij}(t) = 0, T_{ij}(t) = 1) + \Pr(D_{ij}(t) = 1, T_{ij}(t) = 0) \quad (3.33)$$

Les équations 3.25, 3.26, 3.27, 3.28, 3.29 et 3.30 sont également utilisées pour calculer cette probabilité.

3.4.3 Détection basée sur les accusés de réception du protocole de routage

Certains protocoles, tels que CORE, déterminent une partie de la réputation à partir des accusés de réception compris dans le protocole de routage. Si un accusé réception est reçu, tous les nœuds qui étaient supposés acheminer le paquet ont coopéré. Il est donc possible d'augmenter leur réputation en conséquence. Sinon, au moins un de ces nœuds a jeté le paquet.

Si le nœud i fait transiter un paquet par l'entremise du nœud j vers k, la probabilité que le paquet se rende est égale à :

$$\Pr(T_{ij}(t) = 0, T_{jk}(t) = 0) \quad (3.34)$$

Si on considère que le paquet s'est rendu au nœud k, la probabilité que le nœud i reçoive un accusé de réception de la part du nœud k alors que cet accusé réception doit transiter par le nœuds j est quant à elle égale à :

$$\Pr(T_{ji}(t) = 0, T_{kj}(t) = 0) \tag{3.35}$$

La probabilité qu'un accusé réception revienne au nœud i dépend donc de la route que ce paquet a empruntée. On définit une route $\bar{H}$ comme étant une route où les nœuds i et j sont présent sans être nécessairement consécutifs (contrairement à la route définie par l'équation 3.11) et où n autres nœuds sont présents est notée de la façon suivante :

$$\bar{H} = \{k_1, k_2, \ldots, i, k \ldots, j, \ldots, k_n\} \tag{3.36}$$

où le nœud k est le nœud suivant i sur cette route

Par conséquent, la probabilité que i reçoive un accusé réception ayant transité par une route où le nœud j est présent est égale à la probabilité suivante :

$$\Pr(A_{ij\bar{H}} = 1) = \Pr(T_{i,i+1}(t) = 0, T_{k-1,k}(t) = 0, T_{k,k-1}(t) = 1, T_{i+1,i}(t) = 1) \tag{3.37}$$

La variable aléatoire $A_{ij}(t)$ décrit si un accusé de réception est reçu par le nœud i alors qu'un paquet acheminé par le nœud j aurait dû en générer un. Si un accusé de réception a été reçu $A_{ij}(t) = 1$, sinon $A_{ij}(t) = 0$. La probabilité $\Pr(A_{ij}(t) = 1)$ est égale à la proportion des paquets qui passent par une route $\bar{H}$ où $A_{ij\bar{H}} = 1$ et est calculée de la façon suivante :

$$\Pr(A_{ij}(t) = 1) = \frac{\sum_n^{NbH} \gamma_{\bar{H}_n}(t) A_{ij\bar{H}_n}}{\gamma_{ij}(t)} \tag{3.38}$$

où NbH est égal au nombre de routes où j achemine un paquet qui a été préalablement acheminé par i et qui devrait générer un accusé réception.

3.5 Réputation

La réputation est la métrique que les protocoles utilisent afin de classifier les différents nœuds qui composent le réseau. Cette réputation se base sur les différentes méthodes de détections présentées à la section 3.4. La réputation propagée est la

réputation que les nœuds échangent entre eux. Le facteur de confiance détermine si un nœud prend en considération l'avis des autres nœuds afin de bâtir les réputations. La réputation totale est la métrique considérée pour déterminer quelles sont les contres-mesures à mettre en place.

La réputation, peu importe si elle se base sur les observations effectuées par le nœud ou sur celles des autres, est modélisée comme un processus stochastique pouvant prendre des états bien précis. Ces états v sont compris dans l'ensemble V. Cet ensemble est donc défini comme étant l'ensemble des valeurs possibles que peuvent prendre les réputations et est utilisé tout au long de cette section. Ces métriques sont donc décrites. Ensuite, les attaques relatives à la réputation sont définies.

3.5.1 Réputation déduite des observations

La réputation déduite des observations est une métrique qui dépend des observations effectuées par le nœud (TO). Cette réputation est modélisée par un processus stochastique dont les états sont les valeurs que peut prendre la réputation et les taux de changement d'états sont caractérisés par les probabilités $\Pr(TO_{ij}(t) = 0)$ et $\Pr(TO_{ij}(t) = 1)$ et par le nombre de paquets ayant eu à être transmis. La définition du protocole stochastique dépend du protocole utilisé. Voici le processus décrit formellement.

$$\Pr({R^{\text{obs}}}_{ij}(t) = v) = I_3({\bar{R}^{\text{obs}}}_{ij}(t), TO_{ij}(t), \gamma_{ij}(t), v) \tag{3.39}$$

où ${\bar{R}^{\text{obs}}}_{ij}(t)$ correspond à l'ensemble des réputations observées par le passé par le nœud i sur le nœud j, c'est-à-dire

$${\bar{R}^{\text{obs}}}_{ij}(t) = \{{R^{\text{obs}}}_{ij}(0), {R^{\text{obs}}}_{ij}(1), \ldots, {R^{\text{obs}}}_{ij}(t-1)\} \tag{3.40}$$

3.5.2 Réputation déduite des accusés de réceptions

La réputation qui est déduite des accusés de réceptions est une métrique qui dépend du nombre d'accusés réception venant du protocole de routage qui ont été reçus et du nombre de paquets ayant eu à être transmis. Cette réputation, lorsque calculée par le nœud i sur le nœud j est modélisée par la variable aléatoire ${R^{\text{ack}}}_{ij}$.

$$\Pr({R^{\text{ack}}}_{ij}(t) = v) = I_4({\bar{R}^{\text{ack}}}_{ij}(t), A_{ij}(t), \gamma_{ij}(t), v) \tag{3.41}$$

où $\bar{R^{\text{ack}}}_{ij}(t)$ correspond à l'ensemble des réputations basées sur les accusés de réceptions calculées par le passé, c'est-à-dire

$$\bar{R^{\text{ack}}}_{ij}(t) = \{R^{\text{ack}}{}_{ij}(0), R^{\text{ack}}{}_{ij}(1), \ldots, R^{\text{ack}}{}_{ij}(t-1)\} \tag{3.42}$$

3.5.3 Propagation de la réputation

Certains protocoles tels que CONFIDANT prévoient que les nœuds partagent la valeur de la réputation entre eux. Cet échange est modélisé par la variable aléatoire $R^{\text{prop}}{}_{ij}(t)$ indiquant la valeur propagée par i sur j aux autres nœuds

$$\Pr(R^{\text{prop}}{}_{ij}(t) = v) = I_5(\bar{R^{\text{prop}}}_{ij}(t), R^{\text{obs}}{}_{ij}(t), R^{\text{ack}}{}_{ij}(t), R^{\text{tot}}{}_{ij}(t-1), v) \tag{3.43}$$

où $\bar{R^{\text{prop}}}_{ij}(t)$ correspond à l'ensemble des réputations propagées par le passé, c'est-à-dire

$$\bar{R^{\text{prop}}}_{ij}(t) = \{R^{\text{prop}}{}_{ij}(0), R^{\text{prop}}ij(1), \ldots, R^{\text{prop}}{}_{ij}(t-1)\} \tag{3.44}$$

3.5.4 Attaque contre la propagation de la réputation

Lorsqu'une CMUR est utilisée, il est possible pour un attaquant de mentir par rapport à la réputation des autres nœuds. Ainsi, il pourra diffamer un autre nœud ou augmenter la réputation des autres attaquants. Ces deux attaques dépendent de la CMUR qui est utilisée.

Ceci est représenté par l'ensemble de variables aléatoires $J'(t)$ qui représente la réputation qui sera propagée par l'attaquant. Cette variable aléatoire peut donc prendre comme état v compris dans l'ensemble V. Si $J'_{ij}(t) = R^{\text{prop}}{}_{ij}(t)$, le nœud i propage la véritable réputation selon le protocole et les observations qui ont été faites. Sinon, il attaque la propagation de la réputation. Pour les nœuds honnêtes $J'_{ij}(t) = R^{\text{prop}}{}_{ij}(t), \forall t \geq 0, \forall j \in N$.

Cette variable aléatoire a comme fonction de densité de probabilité :

$$\Pr(J'_{ij}(t) = v_1 | R^{\text{prop}}{}_{ij}(t) = v_2) = p^{rep}{}_{ij}(t, v_1, v_2) \tag{3.45}$$

où $v_1, v_2 \in V$ et $p^{rep}{}_{ij}(t)$ est un paramètre de l'attaque.

3.5.5 Facteur de confiance

Lorsque deux nœuds partagent leur réputation à propos d'un tiers, ils vérifient s'ils sont du même avis à l'aide du facteur de confiance. S'ils se font confiance, ils tiendront compte de l'avis de leur voisin afin de calculer les réputations.

Le facteur de confiance est un facteur de pondération propre à chaque CMUR qui prend en compte les réputations propagées par les nœuds et qui sera utilisé dans le calcul de la réputation totale R^{tot} décrit plus bas. Deux nœuds, i et j, calculent ce facteur en comparant les réputations qu'ils donnent à un ensemble de nœuds. Plus les réputations données aux nœuds compris dans cet ensemble est similaire, plus la confiance sera élevée entre i et j et meilleur sera le facteur de confiance. Voici donc la définition du facteur de confiance entre les nœuds i et j. Si le nœud i reçoit les réputations des nœuds $k_1, k_2, \ldots k_n$ en provenance du nœud j,

$$\Pr({f^{\text{c}}}_{ij}(t) = v) = I_6({R^{\text{prop}}}_{ik_1}(t), \ldots, {R^{\text{prop}}}_{ik_n}(t), {R^{\text{prop}}}_{jk_1}, \ldots, {R^{\text{prop}}}_{jk_n}, {R^{\text{tot}}}_{ij}(t), v) \tag{3.46}$$

où $v \in V$, V étant l'ensemble des valeurs pouvant être prises par le facteur de confiance.

3.5.6 Réputation totale

La réputation totale est la métrique qui sera considérée par les CMUR afin de déterminer si un nœud est un attaquant ou non et, par conséquent, quel service doit être rendu à un certain nœud. Cette métrique est également utilisée afin de déterminer quelle sont les meilleures routes vers une destination donnée. Cette réputation prend en compte tant la réputation décrite par les autres nœuds que celle qui est calculée par le nœud observateur.

Si le nœud i reçoit la réputation du nœud j considérée par l'ensemble des nœuds $\{x_1 \ldots x_n \in N \mid x_1 \ldots x_n \neq i, j\}$ la réputation totale du nœud j calculée par i est égale à

$\Pr({R^{\text{tot}}}_{ij}(t) = u) =$

$$I_7({\bar{R}^{\text{tot}}}_{ij}(t), {R^{\text{obs}}}_{ij}(t), {R^{\text{ack}}}_{ij}(t), {R^{\text{prop}}}_{x_1 j}(t), \ldots, {R^{\text{prop}}}_{x_n j}(t), {f^{\text{c}}}_{ix_1}(t), \ldots, {f^{\text{c}}}_{ix_n}(t), u) \tag{3.47}$$

L'ensemble des réputations globales calculées par le passé, $\bar{R^{\text{tot}}}_{ij}(t)$, est défini par

$$\bar{R^{\text{tot}}}_{ij}(t) = \{R^{\text{tot}}{}_{ij}(0), R^{\text{tot}}{}_{ij}(1), \ldots, R^{\text{tot}}{}_{ij}(t-1)\} \tag{3.48}$$

3.6 Indices de performance

Cette section décrit les indices de performance pertinents dans l'analyse de performance des CMUR. Les taux de faux positifs, de faux négatifs, les probabilités de pertes de paquets pour les nœuds honnêtes et malicieux permettent de connaître l'efficacité des différents protocoles de routage utilisant la réputation. Il est important de noter que les valeurs moyennes des indices de performance ne sont pas utilisées étant donné qu'elles ont tendance à donner une fausse impression de l'efficacité d'une CMUR. En effet, dépendamment du réseau, un nœud pourrait avoir un taux de faux positifs plus élevé que les autres dans ses observations. Aussi, seul un nœud pourrait être visé par une attaque de déni de service. De la même façon, certains nœuds qui seraient utilisés comme authorité de certification devraient maintenir des taux de faux positifs et de faux négatifs beaucoup plus bas que les autres nœuds. Voici donc une description formelle de ces différents indices de performance, calculés pour un nœud i.

Faux positifs. Le taux de faux positifs est la proportion de nœuds n'ayant jamais jeté illégalement de paquets ou menti par rapport à la réputation des autres nœuds qui ont au temps t une réputation inférieure à un certain seuil. Ce seuil pourrait, par exemple, causer l'exclusion du RAHM de ce nœud honnête.

La variable NH définie si un nœud est honnête ou non. Si un nœud i est honnête, c'est-à-dire qu'il n'a jamais jeté de paquet malicieusement ni propagé de réputation fausse, $NH_i = 1$, sinon $NH_i = 0$

La variable aléatoire $FP_{ij}(t)$ décrit si la réputation du nœud j calculée par le nœud i au temps t est inférieure à un certain seuil alors que le nœud j est honnête. Si la réputation est inférieure à ce seuil, alors $FP_{ij}(t) = 1$, sinon $FP_{ij}(t) = 0$. La probabilité d'avoir un faux positif est la suivante :

$$\Pr(FP_{ij}(t) = 1) = \Pr(R^{\text{tot}}{}_{ij}(t) < \text{seuil} | NH_j = 1) \tag{3.49}$$

Le taux de faux positifs $FP_i(t)$ au temps t pour le nœud i sera donc égal à la

proportion de nœud honnêtes étant considérés comme étant malicieux par ce nœud, c'est-à-dire :

$$FP_i(t) = \frac{\sum_{j|NH_j=1} FP_{ij}(t)}{NBNH} \tag{3.50}$$

où $NBNH$ est égal au nombre de nœuds honnêtes.

Faux négatifs. Le taux de faux négatifs est la probabilité qu'un nœud ayant au moins une fois jeté des paquets dans un dessein malicieux ou ayant propagée au moins une fois une réputation faussée ait une réputation supérieure à un certain seuil. Ce nœud malicieux aurait donc accès au RAHM sans y avoir droit.

La variable aléatoire FN_{ij} décrit si la réputation du nœud j telle que calculée par le nœud i est supérieure à un seuil alors que le nœud j est malicieux, c'est-à-dire que $NH_j = 0$. La probabilité d'avoir un faux négatif est la suivante :

$$\Pr(FN_{ij}(t) = 1) = \Pr({R^{\text{tot}}}_{ij}(t) > \text{seuil} | NH_j = 0) \tag{3.51}$$

Le taux de faux négatifs $FN_i(t)$ au temps t calculé par le nœud i sera donc égal à la proportion de nœuds malicieux considérés comme honnêtes :

$$FN_i(t) = \frac{\sum_{j|NH_j=0} FN_j(t)}{N - NBNH} \tag{3.52}$$

où $N - NBNH$ est égal au nombre de nœuds malicieux.

Taux de pertes de paquets, nœuds honnêtes. Le taux de paquets perdus par le nœud i, noté $TPNH_i$ alors que ce nœud est honnête permet de quantifier la qualité de service des nœuds honnêtes et donc de mesurer l'efficacité d'une CMUR à protéger celle-ci. Voici ce taux :

$$TPNH_i(t) = \frac{\sum_j \Pr(T_{ij}(t) = 1 | NH_i = 1)\gamma_{ij}(t)}{\sum_j \gamma_{ij}(t)} \tag{3.53}$$

Taux de pertes de paquets, nœuds malicieux. Cette métrique est la proportion de paquets jetés venant d'un nœud malicieux. Si ce taux est beaucoup plus élevé que le taux de pertes de paquets des nœuds honnêtes, cela veut dire qu'un nœud égoïste n'aura aucun avantage à jeter des paquets.

$$TPNM_i(t) = \frac{\sum_j \Pr(T_{ij}(t) = 1 | NH_i = 0)\gamma_{ij}(t)}{\sum_j \gamma_{ij}(t)} \tag{3.54}$$

Il ne faut pas perdre de vue que le but premier des CMUR est de minimiser le taux de pertes de paquets des nœuds honnêtes. Les objectifs secondaires sont de diminuer le nombre de faux positifs et de faux négatifs tout en maximisant le nombre de paquets perdus par les nœuds malicieux. Ce dernier objectif a pour but de décourager les éventuels malfaiteurs.

Le but de l'attaquant sera quant à lui de déterminer quelle seront les matrices J' et J qui maximisent le nombre de faux positifs, de faux négatifs et surtout le taux de pertes de paquets des nœuds honnêtes. Ils voudront parfois maintenir le taux de pertes des nœuds malicieux au dessous d'un certain seuil, dépendamment de leur but.

3.7 Instantiation du modèle

Tel que présenté, le modèle contient encore de nombreuses fonctions qui ne sont pas définies. Elles doivent être adaptées pour chaque configuration qui est modélisée. Une fois qu'elles sont adaptées, elles permettent de tirer certaines conclusions concernant l'efficacité des CMUR.

Dans la revue de littérature, quatre CMUR sont présentées. CORE et CONFIDANT ont plusieurs structures internes qui sont peu définies et qui laisse place à interprétation : par exemple, pour CORE, les fonctions qui gère font varier l'importance des observations en fonction du temps ne sont pas définies. Tout ce qui est certain, c'est qu'une importance plus grande est accordée aux observations faites dans le passé. Pour ce qui est de CONFIDANT, les différentes entités fonctionnelles ne sont pas toutes définies au niveau algorithmique ou mathématique. De la même façon que ces deux CMUR sont très difficilement implémentables avec les informations qui sont données. Il est donc difficile de les modéliser mathématiquement.

Par contre, les CMUR Routeguard et Watchdog sont très bien définies. Watchdog n'est pas présentée car elle est très semblable à Routeguard mais moins complète. Voici donc la description de Routeguard.

3.7.1 Routeguard

La CMUR Routeguard est décrite par les auteurs comme étant très prometteuse et capable d'exclure rapidement les attaquants tout en maintenant la qualité de service des autres nœuds. Dans ce qui suit, les différentes métriques reliées aux probabi-

lités sont décrites formellement. Les accusés réception ne sont pas utilisés et que la réputation n'est pas propagée. Par conséquent, les métriques qui s'y rapportent ne sont pas décrites.

Valeurs des réputations

La réputation observée est représentée par un processus stochastique comprenant quatre variables.

1. La première variable (*v1*) est l'état catégorisant l'action que les autres nœuds prendront vis-à-vis ce nœud. Elle peut prendre les valeurs N (nouveau), Me (Membre), I (Instable), S (Suspect) ou Ma (Malicieux).
2. La seconde (*v2*) représente le niveau de collaboration du nœud. Elle varie en fonction de TO et de la première variable entre les valeurs R_{min} et R_{max}, deux constantes propres à chaque implantation de Routeguard qui représentent la valeur minimale et maximale que peut prendre *v2*.
3. La troisième (*v3*) variable compte le nombre de fois qu'un nœud passe dans l'état Instable à partir de l'état Membre.
4. La quatrième (*v4*) est un compteur qui limite le temps que doit passer la réputation dans un état déterminé.

Poids des liens

Le poids des arcs est défini de la façon suivante

$$W_{ij} = \begin{cases} 1 + |N|E_{ij} & \text{si } {R^{\text{tot}}}_{ij}(t) \neq Ma \\ \infty & \text{sinon} \end{cases} \tag{3.55}$$

Probabilité de jeter un paquet pour réputation trop basse

Cette métrique mesure la probabilité qu'un paquet soit jeté étant donné que la réputation d'un nœud présent sur la route soit trop basse. Pour Routeguard, un paquet sera jeté si le nœud source n'était pas dans l'état membre ou si un nœud intermédiare était dans l'état suspect ou malicieux.

$$\Pr(F_{ij}(t) = 0) = \begin{cases} 1 & \text{si} \quad {R^{\text{tot}}}_{ij}(t) = Me \\ 0 & \text{sinon} \end{cases} \tag{3.56}$$

Réputation totale

La réputation qui est prise en compte n'est composée que de la réputation observée. Ces deux réputations sont donc égales.

$${R^{\text{tot}}}_{ij}(t) = {R^{\text{obs}}}_{ij}(t) \tag{3.57}$$

Réputation observée

Afin de faciliter la lecture, des substitutions ont été effectuées. Ces substitutions sont les suivantes :

$$\gamma = \gamma_{ij}(t)$$

$$v1 = v1(t)$$

$$v2 = v2(t)$$

$$v3 = v3(t)$$

$$v4 = v4(t)$$

$$v1_{t-1} = v1(t-1)$$

$$v2_{t-1} = v2(t-1)$$

$$v3_{t-1} = v3(t-1)$$

$$v4_{t-1} = v4(t-1)$$

Voici le détail du calcul de la réputation observée dans le cas de Routeguard :

$$
{R^{\text{obs}}}_{ij}(t) =
$$

$$
= \begin{cases}
(N, 0, 0, 0) & \text{si} \quad t = 0 \\
(N, v2_{t-1} - 5\gamma, 0, v4_{t-1} + 1) & \text{si} \quad v1_{t-1} = N, v4 < T_N, TO(t) = 1 \\
(N, v2_{t-1} + \gamma, 0, v4_{t-1} + 1) & \text{si} \quad v1_{t-1} = N, v4 < T_N, TO(t) = 0 \\
(S, 0, 0, 0) & \text{si} \quad v1_{t-1} = N, v4 = T_N, v2 < 0 \\
(Me, 0, 0, 0) & \text{si} \quad v1_{t-1} = N, v4 = T_N, v2 > 0 \\
(S, 0, 0, v4_{t-1} + 1) & \text{si} \quad v1 = S, v4 < T_M \\
(S, 0, 0, v4_{t-1} + 1) & \text{si} \quad v1 = S, T_M < v4 < 1,5T_M \\
 & \text{et} \ TO(t) = 0 \\
(Ma, 0, 0, 0) & \text{si} \quad v1 = S, T_M < v4 < 1,5T_M \\
 & \text{et} \quad TO(t) = 1 \\
(I, 0, 0, 0) & \text{si} \quad v1 = S, v4 = 1,5T_M, TO(t) = 0 \\
(Ma, 0, 0, 0) & \text{si} \quad v1 = Sv4 > 1,5T_M \\
 & \text{et} \quad TO(t) = 1 \\
(I, v2_{t-1} - 6\gamma, v3_{t-1}, v4_{t-1} + 1) & \text{si} \quad v1_{t-1} = I, v4 < T_I \\
 & \text{et} \quad TO(t) = 1, v3 < Malcount \\
(I, v2_{t-1} + \gamma, v3_{t-1}, v4_{t-1} + 1) & \text{si} \quad v1_{t-1} = I, v4 < T_I \\
 & \text{et} \quad TO(t) = 0, v3 < Malcount \\
(S, 0, 0, 0) & \text{si} \quad v1_{t-1} = I, v4 = T_I, v2 < 0 \\
(Me, 0, v3_{t-1} + 1, 0) & \text{si} \quad v1_{t-1} = I, v4 = T_I \\
 & \text{et} \quad v2 > 0, v3 < Malcount \\
(Me, v2_{t-1} - 5\gamma, v3_{t-1}, 0) & \text{si} \quad v1_{t-1} = M \\
 & \text{et} \quad v2 > R_{min} \\
(Me, \text{Min}(R_{max}, v2_{t-1} + \gamma), v3_{t-1}, 0) & \text{si} \quad v1_{t-1} = M \\
(I, 0, v3_{t-1} + 1, 0) & \text{si} \quad v1_{t-1} = Me, v2 < R_{min} \\
 & \text{et} \quad v3 < Malcount \\
(S, 0, 0, 0) & \text{si} \quad v1_{t-1} = Me, v2 < R_{min} \\
 & \text{et} \quad v3 > Malcount
\end{cases}
\tag{3.58}
$$

3.8 Analyse du modèle

Le modèle proposé permet de décrire les différents aspects d'un RAHM tels que la demande, la topologie, les CMUR utilisées, les attaquants et leur influence sur différentes métriques comme le taux de pertes de paquets et la probabilité d'observer une telle perte de paquets. Les CMUR se basent sur ces derniers critères afin de construire les réputations. Différents indices de performance à être utilisés pour analyser les CMUR sont également intégrés au sein du modèle. Aucun autre modèle mathématique ne permet de décrire de façon formelle un RAHM.

Il est important de noter qu'il est impossible de résoudre façon analytique le modèle et de calculer les différentes probabilités. En effet, les processus stochastiques qui y sont présents sont des processus stochastiques avec mémoire et les probabilités de transition d'états varient avec le temps. De plus, le calcul de la probabilité $\Pr(E = 1)$ (équation 3.23) est un système d'équations non-linéaire, ce qui empêche également la résolution du modèle. Des méthodes stochastiques doivent donc être utilisées afin d'obtenir des résultats.

Aussi, on peut constater qu'il est impossible d'établir une relation simple entre l'environnement, les attaquants, la demande et les CMUR, qui permettraient de quantifier de façon globale la validité d'une CMUR ou même de trouver des valeurs pour les probabilités décrites au long du chapitre. En effet, il est nécessaire de résoudre le modèle pour chaque implémentation et chaque scénario, une très petite variation dans la configuration entraînant un changement radical des différentes probabilités et donc des valeurs des indices de performances du modèle.

Pour illustrer ce phénomène, on considère deux scénarios où les nœuds i, j et k sont placés dans l'ordre sur une ligne et que le nœud i ne peut communiquer qu'avec le nœud j que le nœud k ne peut communiquer qu'avec le nœud j. Considérons également qu'il n'y a aucun attaquant dans les deux scénarios. Considérons également que la demande moyenne est la même, c'est-à-dire que pour le premier scénario,

$$\Gamma_{ij}(t) = \Gamma_{kj}(t) = c$$

et que pour le second

$$\Gamma_{ji}(t) = \Gamma_{jk}(t) = c$$

Bien que toutes les valeurs moyennes soient les mêmes les valeurs de probabilités

d'échec dues aux collisions seront totalement différentes. En effet, dans le premier scénario, pour i et j

$$\begin{aligned} {\Lambda^{\mathrm{b}}}_{ij} &= \gamma_{jk}(t)T_{jk}(t) && (3.59)\\ \Pr({S^{\mathrm{b}}}_{ij} = 0) &= e^{-{\Lambda^{\mathrm{b}}}_{ij}(T_{\mathrm{ACK}}+T_{\mathrm{DIFS}})} && (3.60)\\ {\Lambda^{\mathrm{d}}}_{ij} &= \gamma_{jk}(t)T_{jk}(t) && (3.61)\\ \Pr({S^{\mathrm{a}}}_{ij} = 0) &= e^{-{\Lambda^{\mathrm{d}}}_{ij}(T_{\mathrm{DATA}}+T_{\mathrm{ACK}})} && (3.62) \end{aligned}$$

tandis que dans le second,

$$\begin{aligned} {\Lambda^{\mathrm{b}}}_{ji} &= 0 && (3.63)\\ \Pr({S^{\mathrm{b}}}_{ij} = 0) &= 1 && (3.64)\\ {\Lambda^{\mathrm{d}}}_{ji} &= 0 && (3.65)\\ \Pr({S^{\mathrm{a}}}_{ij} = 0) &= 1 && (3.66) \end{aligned}$$

Les calculs sont les mêmes pour les nœuds j et k. Dans le premier cas, la probabilité d'échec due aux collisions dépend de la demande tandis que dans le second cas, si le total des flots sortant du nœud j sont inférieurs à la capacité du canal, il ne devrait pas y avoir de collisions, ce qui est tout à fait logique. En effet, dans le second cas, un seul nœud accède au canal. Il ne devrait donc pas entrer en collision avec lui-même, car par hypothèse, le flot total sortant de ce nœud ne dépasse pas la capacité du canal. La simple inversion de la demande fait que la probabilité d'échec passe de 0 à une valeur qui dépend de la demande. On voit donc qu'on ne peut pas utiliser les valeurs moyennes afin de calculer des valeurs moyennes pour les probabilités.

Lors du prochain chapitre, une implémentation du modèle en C++ est décrite. Les résultats obtenus à l'aide de cette implémentation sont validées en les comparant à des résultats obtenus à l'aide du simulateur Qualnet. L'implémentations et de simulations y sont donc discutés en détails, tout comme les méthodes utilisées pour valider statistiquement le modèle.

CHAPITRE 4

Validation du modèle et résultats

Le modèle proposé au chapitre précédant modélise les différents aspects d'un RAHM. La demande, les flots, les attaquants et les CMUR y sont décrits mathématiquement afin de prédire le comportement du système dans son ensemble. Toutefois il est primordial de vérifier si les prédictions qu'il permet de faire sont justes et correspondent à la réalité. Des simulations effectuées à l'aide du logiciel Qualnet sont comparées à des simulations de Monte Carlo.

En effet, le modèle proposé est trop complexe pour être résolu analytiquement. Il faut donc absolument utiliser des méthodes numériques afin de déterminer les valeurs des différentes probabilités.

Ce chapitre présente donc la validation du modèle précédemment décrit. Celle-ci est effectuée en comparant les résultats obtenus à l'aide du modèle et ceux d'un simulateur commercial en simulant un ensemble de scénarios de test. Pour chacun d'entre eux, il sera donc possible de voir les prédictions établies par le modèle ainsi que les résultats qui serviront de référence à la validation. La méthodologie de validation est donc présentée en un premier temps.

Afin de pouvoir implanter le modèle, différents ajustements lui ont été portés. Ceux-ci sont exposés et justifiés dans un premier temps. Ensuite, l'implémentation qui a été faite en C++ est commentée. Les algorithmes qui ont été implantés sont présentés en pseudo-code afin de montrer exactement comment les équations du modèle ont pu être utilisées afin d'obtenir les résultats. Les simulations effectuées à l'aide du simulateur Qualnet sont elles aussi détaillées. Après avoir décrit brièvement le logiciel en question, les paramètres qui sont nécessaires à la simulation sont présentés. Le plan d'expérience est ensuite décrit. Celui-ci décrit quelles sont les expériences qui devront être effectuées afin d'obtenir des résultats reproductibles et statistiquement significatifs. Par après, les métriques qui sont mesurées lors des simulations du modèles sont validés en les comparant aux métriques obtenues en utilisant Qualnet. Le test statistique de Kolmogorov-Smirnov permet de voir si les résultats correspondent

ou non. Une analyse des différents résultats est ensuite effectuée, tant pour déterminer de l'efficacité du modèle que pour mesurer l'efficacité de certaines stratégies utilisées dans les CMUR.

4.1 Méthodologie de validation

Afin de déterminer si le modèle est fiable, il est nécessaire de vérifier si les résultats que celui-ci donne concordent avec la réalité. Comme il est difficile de réaliser des expériences sur un vrai RAHM, des résultats provenant d'un simulateur reconnus seront utilisés comme résultats de référence.

Pour obtenir les résultats à partir du modèle, il serait nécessaire de le résoudre analytiquement, ce qui est difficile. En effet, la plupart des processus stochastiques ne sont pas modélisable par une chaîne de Markov et sont des processus avec mémoire. Il est donc impossible d'obtenir une matrice de transition constante qui permettrait de trouver les différentes probabilités après un certain temps. Aussi, plusieurs équations ne sont pas résolvables de façon exacte, comme par exemple l'équation 3.23, qui correspond à un système d'équations non-linéaires qui ne peut pas être résolu de façon analytique.

Toutes ces raisons font qu'il est nécessaire d'utiliser des méthodes stochastiques, plus précisément des méthodes de Monte Carlo, et d'effectuer certaines simplifications qui permettent de résoudre les systèmes d'équations non-linéaires.

Avec l'implantation du modèle et de Qualnet, il est possible de réaliser le plan d'expérience tel que décrit plus bas.

Des scénarios sont créés à l'aide d'une application codée en C++. Celle-ci génère aléatoirement les positions, la demande entre chaque nœuds et la grandeur du terrain. Elle détermine également quels nœuds sont des attaquants et quelle quantité de paquets ceux-ci jettent. Toutes ces données sont fournies en paramètres à l'application en respectant le plan d'expérience.

Une fois que les scénarios sont créés, ceux-ci sont traités indépendamment à l'aide de Qualnet et du modèle. Les résultats relatifs aux indices de performances établis dans le plan d'expérience sont par la suite comparés et analysés à l'aide de tests statistiques.

4.2 Adaptation du modèle

Le modèle proposé doit être adapté à chaque problème devant être simulé. Cette section décrit les différentes fonctions qui ont été implantées dans les simulations de Monte Carlo et qui ont servi à adapter les protocoles compris dans le logiciel Qualnet.

Tout d'abord, le trafic est défini de façon formelle. Ensuite, la mobilité et la topologie utilisée est décrite. Par la suite, le protocole de routage utilisé est présenté, tout comme les CMUR et les attaquants. Finalement, les différentes fonctions décrivant la réputation et la détection sont décrites.

Connectivité. Le calcul de la connectivité prend en compte plusieurs facteurs qui ne sont pas connus lors des simulations. En effet, Qualnet ne permet pas de connaître à priori la puissance du signal à un mètre de l'antenne. Ensuite, la variable aléatoire VAA n'est pas connue. Pour contourner ces problèmes, l'application `Radio_Range`, fournie avec la suite Qualnet, a été utilisée afin de connaître la portée moyenne dans le réseau simulé avec les paramètres de simulation utilisés. Cette distance est égale à 65 m. Toutefois, lors des expériences, la portée maximale s'est toujours révélée légèrement supérieure à 65 m. De plus, la portée maximale varie légèrement, tel que décrit dans le modèle. Cette variation n'est toutefois pas décrite par l'application `Radio_Range`. Il a donc fallu déterminer de façon empirique les paramètres de la variable aléatoire VAA.

Afin d'obtenir des valeurs appropriées pour la variance $STDA$ de la variable aléatoire VAA' et L' de l'équation 3.4, 20 scénarios ont été simulés 20 fois chacun. Ces scénarios comprenaient deux nœuds à une distance variant entre 65 et 80 mètres. Pour chacune des relances, on a vérifié si les deux nœuds étaient capables de commnuniquer entre eux. Si oui, on affectait la valeur 1 à une variable aléatoire C^{Qualnet}. Sinon, on affectait la valeur 0 à cette variable aléatoire. Il a donc été possible de calculer la probabilité que $C^{\text{Qualnet}} = 1$ en fonction de la distance, c'est-à-dire

$$\Pr(C_{ij}^{\text{Qualnet}} = 1) = f(STDA, L'_{ij}))$$

L'équation 3.4 a été codée dans Matlab afin de calculer la probabilité $\Pr(C_{ij}^{\text{Matlab}} = 1)$. Différentes valeurs ont été données aux paramètres $STDA$ et L'. Pour chacune de ces valeurs, la probabilité $\Pr(C_{ij}^{\text{Matlab}} = 1)$ a été calculée.

Les probabilités calculées à l'aide de Qualnet et de Matlab ont par la suite été comparées à l'aide du test de Kolmogorov-Smirnov, décrit à la section 4.6.2. Lorsque le test a pu déterminer que les deux distributions venaient de la même variable aléatoire, les paramètres optimaux ont été obtenus. Ceux-ci sont égaux à $STDA' = 0,1$ et $L' = 66,7m$.

Topologie. Aucune mobilité ne sera supportée. La position des nœuds est fixe dans le temps

$$L_{ij}(t) = l_{ij}, \forall t \tag{4.1}$$

où l_{ij} est un paramètre de simulation.

Cette simplification permet de réduire la complexité de l'implémentation du modèle. Toutefois, elle fait que l'analyse de performance ne sera valide que pour des scénarios où la mobilité est faible. Lors de travaux futurs, il serait important de valider le modèle avec des scénarios où la mobilité est importante.

Protocole de routage. Le protocole de routage utilisé est DSR. La recherche de route est modélisée par un algorithme de recherche de plus court chemin. L'algorithme du calcul des routes est donc basé sur l'algorithme de Dijkstra. La principale différence est que le poids des arcs est égal à $1+|N|E_{ij}(t)$ si $C_{ij}(t) = 1$ et égal à 0 sinon.

Calcul de la détection. Les erreurs de détection ne peuvent se produire que pour les paquets dont la route contient au minimum 4 nœuds. En effet, si on considère l'exemple où le nœud i écoute pour déterminer si un nœud j achemine le paquet au nœud k, un nœud q ne pourra fausser l'observation du nœud i que s'il n'est pas à portée des nœuds j et k. Comme les scénarios utilisés ne comprennent que 20 nœuds, que la grandeur du terrain est au maximum de 400 m et que la portée des nœuds est d'environ 60 m, il est très peu probable que les graphes soient d'un diamètre suffisant. La probabilité $\Pr(D_{ij}(t) = 0)$ (équation 3.29) est donc fixée à zéro.

$$\Pr(D_{ij}(t) = 0) = 0$$

CMUR. Aucune contre-mesure utilisant la réputation ne sera utilisée. L'analyse de performance ne permet pas de montrer à elle seule que le modèle est adéquat pour simuler les CMUR présentes dans les RAHM. Toutefois, les calculs de réputations se basent sur les probabilités $\Pr(T_{ij}(t) = 1)$, $\Pr(TO_{ij}(t) = 1)$. Si le calcul de ces dernières est adéquat, le calcul des réputations devrait l'être

également. Il sera bon de valider cette hypothèse lors de travaux futurs. Il est toutefois évident que lorsque le protocole DSR est utilisé, l'analyse de performance est tout à fait adéquate, sauf dans le cas où la mobilité est très grande.

Attaquants. La probabilité qu'un nœud jette un paquet est constante dans le temps. Vu que les CMUR ne sont pas prises en compte, la probabilité qu'un attaquant en aide un autre en lui attribuant une bonne réputation ou qu'il diffame un nœud honnête est nulle.

$$\Pr(J_{ij}(t) = 1) = \eta_{ij}, 0 \leq \eta_{ij} \leq 1 \tag{4.2}$$

$$\Pr(J'_{ij}(t) = R^{\text{prop}}{}_{ij}(t)) = 1 \tag{4.3}$$

où η_{ij} est un paramètre de simulation.

Accusés de réception. Comme le protocole DSR est utilisé, la probabilité d'obtenir un accusé réception est nulle. En effet, aucun accusé de réception de ce type n'est compris dans ce protocole.

$$\Pr(A_{ij}(t) = 0) = 1 \tag{4.4}$$

4.3 Implémentation du modèle

Comme il a été énoncé précédemment, le modèle proposé est non-markovien. Les différentes métriques doivent être obtenues à l'aide de simulations. Les méthodes de Monte-Carlo ont été choisies pour calculer les différentes valeurs des probabilités. Ces simulations ont été codées en C++ car ce language est très performant et son aspect orienté-objet facilite la réutilisation du code.

Tout d'abord, une simplification qui a été apportée au modèle afin de faciliter son implantation est explicitée. Cette section décrit ensuite de quelle façon les principales fonctions ont été traduites en C++ en présentant leur algorithme. L'algorithme de haut niveau ainsi que l'algorithme du calcul des routes sont explicités. Pour chacune de ces fonctions, la complexité algorithmique est ensuite calculée afin de prédire l'évolution de la performance en fonction du nombre de nœuds considérés.

Résolution des systèmes non-linéaires. Afin de pouvoir simuler les différents systèmes d'équations non-linéaires, une simplification est effectuée. Les équa-

tions dépendent du moment précédent afin d'éviter les références circulaires. Par exemple, pour le calcul des interférences, l'équation

$$\Lambda^{a}{}_{i}(t) = \sum_{k|k\neq i, C_{ki}(t)=1} \sum_{x} (1 - S^{a}{}_{kx}(t))(1 - S^{b}{}_{kx}(t))(1 - J_{kx}(t))(1 - F_{kx}(t))\gamma_{kx}(t)$$

sera simplifiée par l'équation

$\Lambda^{a}{}_{i}(t) =$

$$\sum_{k|k\neq i, C_{ki}(t)=1} \sum_{x} (1-S^{a}{}_{kx}(t-1))(1-S^{b}{}_{kx}(t-1))(1-J_{kx}(t-1))(1-F_{kx}(t-1))\gamma_{kx}(t)$$

Un calcul équivalent est fait pour résoudre les autres systèmes d'équations. Ceci est valide lorsque les différentes métriques ne varient pas très brusquement dans le temps.

Algorithme de haut niveau. Le modèle fonctionne en calculant les états des variables pour chaque incrément de temps pour ensuite calculer les valeurs moyennes des métriques pour chaque paire de nœuds. L'intervalle de temps est égal à 1 s. Plusieurs sessions sont exécutées avec des graines d'initialisation différentes. La variable *Seuil* présente dans l'algorithme représente l'écart maximal accepté entre deux itérations pour les valeurs cumulatives. C'est cette variable qui détermine la précision du modèle. Il est bon de noter qu'une précision trop grande peut faire que le modèle ne converge jamais. C'est pourquoi un critère d'arrêt est considéré. Si le temps dépasse une valeur, décrite dans l'algorithme par *Critere_Arret*, le programme arrête et retourne les résultats obtenus.

Calcul des routes. Le calcul de la matrice $\gamma_{ij}(t)$ et des routes est basé sur l'algorithme de Dijkstra. Cet algorithme calcule les plus courts chemins à partir d'un nœud. Dans le pire cas, chaque nœud calcule la valeur de la distance avec le nœud source pour $n-1$ nœuds. Cette opération est répétée dans le pire cas au maximum n fois.

Cet algorithme étant répété n fois, la complexité algorithmique, pour le pire cas, est donc égale à

$$O(n^3)$$

Étant donné que cet algorithme est très connu, il n'est pas présenté.

1: Lire fichiers de configuration
2: **pour** $i = 0$ à Nb_Sessions **faire**
3: $t \leftarrow 0$
4: Initialiser les matrices cumulatives à 0
5: $Convergeance \leftarrow FALSE$
6: **tant que** $\neg Convergeance \cap Critere\_Arret > t$ **faire**
7: $E\_Cumulatif_{ij}(t) \leftarrow E_{ij}(t) + E\_Cumulatif_{ij}(t-1)$
8: $F\_Cumulatif_{ij}(t) \leftarrow F_{ij}(t) + F\_Cumulatif_{ij}(t-1)$
9: $T\_Cumulatif_{ij}(t) \leftarrow T_{ij}(t) + T\_Cumulatif_{ij}(t-1)$
10: $D\_Cumulatif_{ij}(t) \leftarrow D_{ij}(t) + D\_Cumulatif_{ij}(t-1)$
11: $TO\_Cumulatif_{ij}(t) \leftarrow TO_{ij}(t) + TO\_Cumulatif_{ij}(t-1)$
12: $A\_Cumulatif_{ij}(t) \leftarrow A_{ij}(t) + A\_Cumulatif_{ij}(t-1)$
13: $\gamma\_Cumulatif_{ij}(t) \leftarrow \gamma_{ij}(t) + \gamma\_Cumulatif_{ij}(t-1)$
14: $t \leftarrow t + 1$
15: **si** Pour toutes les matrices cumulatives $|Matrice\_Cumulative_{ij}(t) - Matrice\_Cumulative_{ij}(t-1)| < Seuil$ **alors**
16: $Convergeance \leftarrow TRUE$
17: **sinon**
18: $Convergeance \leftarrow FALSE$
19: **fin si**
20: **fin tant que**
21: $E\_Cumulatif_{ij}(t) \leftarrow \frac{E\_Cumulatif_{ij}(t)}{t}$
22: $F\_Cumulatif_{ij}(t) \leftarrow \frac{F\_Cumulatif_{ij}(t)}{t}$
23: $T\_Cumulatif_{ij}(t) \leftarrow \frac{T\_Cumulatif_{ij}(t)}{t}$
24: $D\_Cumulatif_{ij}(t) \leftarrow \frac{D\_Cumulatif_{ij}(t)}{t}$
25: $TO\_Cumulatif_{ij}(t) \leftarrow \frac{TO\_Cumulatif_{ij}(t)}{t}$
26: $A\_Cumulatif_{ij}(t) \leftarrow \frac{A\_Cumulatif_{ij}(t)}{t}$
27: $\gamma\_Cumulatif_{ij}(t) \leftarrow \frac{\gamma\_Cumulatif_{ij}(t)}{t}$
28: Retourner matrices cumulatives
29: **fin pour**

FIGURE 4.1 Algorithme de haut niveau

Calcul de $\mathbf{E_{ij}(t)}$. La complexité algorithmique de ce calcul représenté par l'algorithme 4.3, en fonction du nombre de nœuds n, est égal dans le pire cas à

$$O(n^4 + n^3) \approx O(n^4)$$

Calcul de $\mathbf{T_{ij}(t)}$. Le calcul de la probabilité qu'un paquet soit jeté par le nœud j est donné par l'algorithme 4.3. On peut voir que la complexité algorithmique est égale à

$$O(n^3 + n^2) \approx O(n^3)$$

Ainsi, la complexité globale d'une itération, c'est-à-dire le passage du temps t au temps $t + 1\text{s}$, est égale à

$$O(n^4)$$

4.4 Simulations Qualnet

Le simulateur Qualnet a été utilisé afin d'obtenir des résultats de référence et de valider ceux obtenus à l'aide des simulations de Monte Carlo. Comme il est peu réaliste de mettre sur pied un véritable RAHM afin d'obtenir les résultats de référence, l'utilisation d'un simulateur était nécessaire.

Dans cette section, le logiciel Qualnet est brièvement décrit. Ensuite, les modifications apportées au code source du logiciel sont énoncées. Finalement, les valeurs utilisées pour configurer les simulations sont présentés.

4.4.1 Description du logiciel Qualnet

Qualnet est un logiciel permettant de simuler différents types de réseaux avec ou sans fils, utilisant de nombreux protocoles basés sur TCP/IP tout en tenant compte de modèles de propagation d'ondes et de mobilité. Ce logiciel est employé largement autant par la communauté scientifique que par l'industrie. Ce simulateur prend en compte l'environnement et les différents événements physiques qui peuvent influencer la propagation des ondes. Ainsi, les conditions météorologiques sont évaluées, tout comme le bruit émis par les nœuds.

1: **pour** $i = 0$ à Nb_Noeuds **faire**
2: **pour** $j = 0$ à Nb_Noeuds **faire**
3: $P\_Interference_{ij} \leftarrow 0$
4: **si** $\gamma_{ij}(t) \bigcap C_{ij}(t)$ **alors**
5: **pour** $k = 0$ àNb_Noeuds **faire**
6: **pour** $l = 0$ àNb_Noeuds **faire**
7: **si** $C_{ki}(t)$ **alors**
8: **si** $\neg T_{kl}$ **alors**
9: $Trafic\_Interference_{ij} \leftarrow Trafic\_Interference_{ij} + \gamma_{kl}(TDIFS + TDATA)$
10: **fin si**
11: **si** $\neg T_{lk}$ **alors**
12: $Trafic\_Interference_{ij} \leftarrow Trafic\_Interference_{ij} + \gamma_{lk}(TDIFS + TACK)$
13: **fin si**
14: **fin si**
15: **si** $\neg C_{ki}(t) \bigcap C_{kj}(t)$ **alors**
16: **si** $\neg T_{kl}$ **alors**
17: $Trafic\_Interference_{ij} \leftarrow Trafic\_Interference_{ij} + \gamma_{kl}(TDATA + TDATA)$
18: **fin si**
19: **si** $\neg T_{lk}$ **alors**
20: $Trafic\_Interference_{ij} \leftarrow Trafic\_Interference_{ij} + \gamma_{lk}(TDATA + TACK)$
21: **fin si**
22: **fin si**
23: **fin pour**
24: **fin pour**
25: $P\_Interference_{ij} \leftarrow 1 - e^{-Trafic\_Interference}$
26: **si** $P\_Interference_{ij} > RAND$ **alors**
27: $E_{ij}(t) \leftarrow FALSE$
28: **sinon**
29: $E_{ij}(t) \leftarrow TRUE$
30: **fin si**
31: **sinon**
32: $E_{ij}(t) \leftarrow FALSE$
33: **fin si**
34: **fin pour**
35: **fin pour**
36: Retourner $E(t)$

FIGURE 4.2 Algorithme du calcul de $\Pr(E_{ij}(t) = 1)$

1: **pour** $i = 0$ à Nb_Noeuds **faire**
2: **pour** $j = 0$ à Nb_Noeuds **faire**
3: **si** $Grandeur\_Route \geq 3$ **alors**
4: **pour** k à $Grandeur\_Route_{i,j}$-2 **faire**
5: $I_2 \leftarrow Route_{i,j}[k], I_3 \leftarrow Route_{i,j}[k+1], I_4 \leftarrow Route_{i,j}[k+2]$
6: $TraficTotal_{I_2,I_3} \leftarrow TraficTotal_{I_2,I_3}(t) + \Gamma_{i,j}(t)$
7: **si** $E_{I_3,I_4} \cup E_{I_2,I_3}$ **alors**
8: $TraficPerdu_{I_2,I_3} \leftarrow TraficPerdu_{I_2,I_3}(t) + \Gamma_{i,j}(t)$
9: **fin si**
10: **si** $p^{jeter}_{I_3,I_4}$ **alors**
11: $TraficMalicieux_{I_2,I_3} \leftarrow TraficMalicieux_{I_2,I_3}(t) + \Gamma_{i,j}(t)$
12: **fin si**
13: **fin pour**
14: **fin si**
15: **si** $Grandeur\_Route = 2$ **alors**
16: $TraficTotal_{i,j} \leftarrow TraficTotal_{i,j}(t) + \Gamma_{i,j}(t)$
17: **si** $E_{i,j}$ **alors**
18: $TraficPerdu_{i,j} \leftarrow TraficPerdu_{i,j}(t) + \Gamma_{i,j}(t)$
19: **fin si**
20: **fin si**
21: **fin pour**
22: **fin pour**
23: **pour** $i = 0$ à Nb_Noeuds **faire**
24: **pour** $j = 0$ à Nb_Noeuds **faire**
25: **si** $Trafic\_Total_{i,j} > 0$ **alors**
26: **si** $\frac{TraficPerdu_{I_2,I_3}}{TraficTotal_{i,j}} > RAND \bigcup \frac{TraficMalicieux_{I_2,I_3}}{TraficTotal_{i,j}} > RAND$ **alors**
27: $T_{i,j}(t) \leftarrow TRUE$
28: **sinon**
29: $T_{i,j}(t) \leftarrow FALSE$
30: **fin si**
31: **sinon**
32: $T_{i,j}(t) \leftarrow FALSE$
33: **fin si**
34: **fin pour**
35: **fin pour**
36: Retourner $T(t)$

FIGURE 4.3 Algorithme du calcul de $\Pr(T_{ij}(t) = 1)$

Comme sa version académique GloMoSim, ce simulateur a été conçu par couches correspondant à celles de la suite de protocoles TCP/IP. Chaque nœud a donc sa propre pile de protocoles. Les différents messages sont passés entre les couches.

Un avantage de Qualnet est qu'il y est possible d'implémenter ses propres protocoles, et ce à n'importe quelle couche du réseau. Ainsi, il a été possible de modifier le protocole DSR afin d'y implémenter un attaquant et de recueillir les différentes statistiques nécessaire à la validation du modèle.

Le language utilisé est le C, autant pour les modules existant que pour ceux qui doivent être créés. Par la suite, il faut compiler l'exécutable `Qualnet.exe` qui est utilisé par les différents modules décrits ci-dessous. Ces modules utilisent quant à eux Java. Voici une courte description des principaux modules du logiciel.

Scenario designer. Ce module sert à créer des scénarios. Il permet l'édition automatisée des différents fichiers de configuration. Il est également possible de déterminer la position initiale des nœuds et de leur attribuer des propriétés. Ainsi, il est possible de leur attribuer l'utilisation de protocoles, d'un modèle de propagation et d'un modèle de déplacement.

Animator. Ce module permet d'exécuter les scénarios qui ont été créés. Il est donc possible de voir l'envoi des paquets d'un nœud à l'autre.

Analyser. Ce module permet d'afficher les statistiques qui ont été récoltées tout au long de la simulation. Par exemple, dans le fichier dsr.cpp, la structure `dsr->stats` entrepose les différentes statistiques.

Protocol Designer. Ce module permet de créer un nouveau protocole. Il crée un squelette qui facilite grandement son implémentation. Il est donc possible d'y créer différentes structures de données et d'y ajouter certaines variables qui définiront le protocole.

Packet Tracer. Ce module permet de conserver des informations concernant les paquets qui sont transmis entre les nœuds. Il est donc possible de suivre toutes les opérations qui sont effectuées sur chacun des paquets à toutes les couches des protocoles. Par contre, il est bon de noter que la vitesse de simulation est grandement diminuée.

Mode console. Il est possible d'exécuter Qualnet à l'aide de la console. Ce module est beaucoup plus rapide que le module *Animator*. De plus, il existe un outil,

`Radio_range.exe`, qui permet de connaître approximativement la portée des nœuds dans un scénario donné.

Ces modules peuvent grandement aider la création et l'exécution des simulations. Toutefois, seul le mode console a été utilisé. En effet, il est impossible de créer aléatoirement des nouveaux scénarios à l'aide de ces modules. Le module *Analyser* permet l'affichage des statistiques, mais dans notre cas, il est préférable d'exporter celles-ci dans des fichiers et de les traiter à l'aide d'un logiciel comme Matlab. L'implémentation du protocole DSR étant fournie avec le logiciel Qualnet et ne devant être que légèrement modifié pour nos besoins, il est inutile d'utiliser le *Protocol Designer*. Finalement, l'utilisation des traces est beaucoup trop lourde pour être utilisée. L'utilisation du mode console est beaucoup plus approprié et permet l'utilisation de fichiers de scripts `.bat` .

4.4.2 Modification des fichiers `.cpp`

Afin de pouvoir récupérer les différentes métriques et d'implémenter un attaquant, il a été nécessaire de modifier le protocole DSR. De plus, il a fallu ajouter des fonctions permettant de mesurer ces métriques. Les fichiers suivants ont donc été modifiés ou ajoutés au sein de Qualnet.

`dsr.cpp`. Ce fichier implémente le protocole DSR. C'est dans ce fichier que sont recueillies les différentes statistiques et où l'attaquant est implanté. Voici les fonctions qui ont dues être modifiées.

`DsrTransmitDataWithSrcRoute`. Cette fonction gère la transmission des paquets utilisant DSR. Cette fonction n'est appelée que par le nœud origine du paquet. Les messages transmis sont mis dans la liste ListeMessagesTO.

`DsrHandlePacketWithSrcRoute`. Cette fonction gère la retransmission des paquets utilisant le protocole DSR. Chaque fois qu'un nœud reçoit un paquet qu'il doit retransmettre, cette fonction est appelée. C'est donc au sein de cette fonction que l'attaquant a été implémenté.

`DsrPeekFunction`. Cette fonction écoute le médium et gère les messages qui ne sont pas destinés au nœud qui écoute. La fonction gérant les accusés de

réception passifs est donc intégrée dans cette fonction. Lorsqu'un paquet est reçu et qu'il est dans la liste des messages envoyés (`ListeMessagesTO`), la probabilité d'observer qu'un paquet a été jeté est diminuée à l'aide de la fonction `ModifierMatriceTO` comprise dans le fichier `StatistiquesMatrices`.

`DsrMacAckHandler`. Cette fonction gère les accusés réception provenant de la couche liaison. Le calcul des pertes de paquets causées par les inferférences y est donc géré à l'aide de la fonction ModifierMatriceE comprise dans le fichier `StatistiquesMatrices`.

`StatistiquesMatrices.cpp`. Ce fichier implémente les fonctions qui compilent les statistiques nécessaires pour pouvoir valider le fonctionnement du modèle. La structure de données `Statistiques` contient les métriques calculées par telles qu'évaluées par un nœud i sur un nœud j. La structure de données `ListeStatistiques` est une liste de structures de données `Statistiques` qui comprend donc les statistiques évaluées par le nœud i sur tous les autres nœuds.

`ListeMessagesTO.cpp`. Ce fichier permet de conserver les messages qui sont envoyés sur le réseau afin de pouvoir calculer le pourcentage de messages qui sont perdus. La structure de données `MessageTO` correspond aux messages qui ont été envoyés par le nœud observateur. La structure de données `ListeMessagesTO` correspond quant à elle à la liste de messages. Chaque nœud utilisant le protocole DSR contient une structure de type `ListeMessagesTO`.

4.4.3 Configuration des simulations

Qualnet est un simulateur qui peut prendre en compte plusieurs paramètres. Ceux-ci influencent grandement son comportement. La plupart des paramètres ont été laissés à leur valeur par défaut. Par contre, le taux de transfert maximal, la demande entre les différents nœuds, la puissance du signal et le gain des antennes sont des paramètres qui se reflètent dans les simulations de Monte Carlo : le taux de transfert maximal détermine quelle demande le réseau pourra acheminer, le flot entre les nœuds est une métrique intégrale au modèle, tandis que la puissance du signal et le gain des antennes détermine quelle sera la portée maximale du signal.

Taux de transfert maximal. Ce taux a été établi à 11 Mbps. Ce taux est le maximum qui peut être établi dans Qualnet. Comme le taux de transfert maximal

est généralement plus élevé dans la plupart des implémentations modernes du standard 802.11, il était logique de prendre le taux maximal offert par Qualnet.

Taille des trames 802.11. La taille de la charge utile des paquets est fixée à 512 octets. La taille de l'entête UDP est égale à 8 octets, et la taille de l'entête IP est égale à 20 octets. La taille de l'entête DSR qui est commune à tous les paquets est égale à 8 octets. L'entête DSR Source Route servant au routage a quant à elle une taille de 8 octets. La taille de l'entête 802.11 est quant à elle égale à 32 octets pour ce qui est des données. La taille totale d'un paquet de données est donc égale à 588 octets. Par conséquent,

$$T_{\text{DATA}} = \frac{588 \times 8}{11} = 428 \quad \mu s$$

La taille d'un accusé réception 802.11 étant égale à 14 octets, $T_{\text{ACK}} = 10 \quad \mu$s, tandis que la durée T_{DIFS} d'un intervalle DIFS est égale à 50 μs

Puissance du signal et gain des antennes. La sensibilité des nœuds a été laissée à la valeur par défaut (-89 dBm). Le gain des antennes a quant à lui été fixé à 1 dB afin que la portée maximale des nœuds soit d'environ 65 m, portée calculée à l'aide de l'application `Radio_Range.exe`. Ce gain a peu d'impact sur la validité du modèle. Choisir une autre valeur aura simplement comme effet que la portée sera différente et que, pour un scénario donné, la matrice d'adjacence sera différente. La matrice d'adjacence nécessaire au fonctionnement du modèle est obtenue à l'aide de Qualnet, ce qui fait que l'impact de ces paramètres est limité.

Demande. La demande est fixé dans le fichier `Qualnet.app`. Cette demande correspond à $\Gamma_{ij}(t)$. Il est fixé aléatoirement tel que décrit à la section 4.5.3.

4.5 Plan d'expérience

Afin d'assurer la validité et la reproductibilité des résultats, il est nécessaire de concevoir un plan d'expérience adéquat. Ce plan doit déterminer quels seront les indices de performances et quels seront les facteurs considérés. Ensuite, il faut déterminer quels seront les combinaisons de facteurs choisies ainsi que le nombre de relances nécessaires afin d'obtenir des données qui seront valides sur le plan statistique.

Dans cette section, le plan d'expérience sera décrit en détail. Tout d'abord, les indices considérés sont explicités. Ensuite, les facteurs considérés sont exposés tout comme leurs niveaux. Finalement, les configurations des simulations qui ont été effectuées sont décrites.

4.5.1 Indices

Afin de déterminer si le modèle est adéquat, il faut être en mesure de vérifier si les métriques sur lesquelles se base le calcul des réputations concordent. Le calcul des probabilités de collisions et des pertes de paquets observées sont très importantes pour le calcul des réputations. En effet, les réputations se basent principalement sur ces métriques. Voici la description des indices qui sont observés lors des simulations.

Probabilité de collision. Une très grande proportion des paquets sont jetés à cause des collisions qui sont présentes dans les réseaux. Il est donc primordial de confirmer la validité du calcul de E (équation 3.23).

Probabilité d'observer la perte d'un paquet. La probabilité TO, décrite par l'équation 3.32 est validée car plusieurs CMUR utilisent les accusés de réception passifs pour le calcul des réputations.

Il est bon de noter que la probabilité qu'un paquet soit jeté, c'est-à-dire $\Pr(T_{ij}(t) = 1)$ (équation 3.26), n'est pas considérée. En effet, ce paramètre n'est pas observable directement dans Qualnet, sauf si un protocole utilise les accusés de réception.

4.5.2 Facteurs

Les facteurs sont des paramètres qui ont une influence sur les indices de performance. Voici les différents facteurs qui seront contrôlés afin de créer les différentes configurations qui serviront à valider le fonctionnement du modèle.

Nombre de nœuds. Ce facteur détermine le nombre de nœuds compris dans le réseau. Ce facteur a un grand impact sur le flot que chaque nœud pourra recevoir ainsi que sur la création des routes.

Demande. Le facteur Γ_{max} détermine la demande qui pourrait être présente entre deux nœuds. Dans la conception des expériences, chaque nœud envoie des mes-

sages à chacun des autres nœuds. La demande entre le nœud i et le nœud j suit une distribution uniforme qui a comme intervalle $[0, \Gamma_{max}]$.

Position des nœuds. Les nœuds sont positionnés aléatoirement dans le réseau. Les coordonnées (x_i, y_i) des nœuds suivent chacune une loi uniforme ayant pour intervalle $[0, L_{max}]$, où le facteur L_{max} détermine la grandeur du terrain.

Probabilité d'être malicieux. Chaque nœud peut être malicieux ou honnête. Le facteur M détermine la probabilité qu'un nœud soit malicieux.

Probabilité de jeter un paquet. Ce facteur détermine la probabilité qu'un attaquant jette un paquet. Le facteur J détermine cette probabilité et est le même pour tous les nœuds malicieux.

4.5.3 Choix des niveaux

Dans la conception d'un plan d'expérience, il est très important de définir quels sont les niveaux significatifs que devraient prendre les facteurs afin d'obtenir des résultats concluants. Il faut par la suite concevoir chacune des expériences en faisant une combinaison de ces facteurs. Cette combinaison doit être faite judicieusement. En effet, faire une combinaison de toutes les valeurs de facteurs entraîne une trop grande quantité d'expériences tandis que trop peu de combinaisons ne permettront pas de valider le fonctionnement du modèle.

Dans le plan d'expérience, chaque combinaison de niveaux de facteurs correspond à une configuration donnée. Pour chaque configuration, plusieurs scénarios sont construits aléatoirement. Les positions de chaque nœud sont donc déterminées, une matrice de demande est générée tout comme les différentes matrices d'attaque. Les scénarios sont alors lancés plusieurs fois avec des graines d'initialisation différentes. Pour une graine d'initialisation donnée, une seule exécution de Qualnet est effectuée tandis que le modèle est simulé jusqu'à ce que les valeurs des indices convergent. Un nombre maximal d'itérations est donné au programme en paramètre pour éviter que son exécution soit trop longue.

Dans ce qui suit, les valeurs significatives des facteurs seront associées afin de faire différentes expériences. Tout d'abord, les valeurs significatives des facteurs sont présentées. Par la suite, plusieurs configurations sont exposées. Les premières visent à valider sommairement le comportement du modèle et son implémentation tandis que les autres visent à vérifier l'adéquation entre le modèle et le simulateur Qualnet.

Valeurs pertinentes

Il est important de déterminer quels sont les niveaux pertinents pour les facteurs afin de valider le modèle. Ces niveaux doivent aussi aider à déterminer dans quelles conditions le modèle est fiable et quelles sont ses limites. Le tableau 4.1 présente ces valeurs.

TABLEAU 4.1 Valeurs pertinentes

Nombre Noeuds (Noeuds)	Trafic (Paquets/s)	Distance (m)	Proportion Attaquants	Paquets Jetés
2	0,5	1	0	0
10	1	100	0,25	0,25
20	5	200	0,5	0,5
30	10	300	0,75	0,75
40	100	400	1	1

Les valeurs attribuées au nombre de nœuds présents dans le réseau varient entre deux nœuds et quarante. Cet intervalle de valeurs a pour but de vérifier quelle est la croissance du temps de calcul et de comparer sur ce critère le modèle avec le logiciel Qualnet. La demande varie quant à elle de 0,5 paquet par seconde à 100 paquets par secondes. Ainsi, il est possible de voir l'impact de la congestion sur la précision du modèle. Les valeurs supérieures à 5 ne seront utilisées que lors de la phase de calibration. La taille du terrain permet quant à elle de vérifier l'impact de la topologie sur la précision. Les tailles du terrain sont choisies afin d'avoir une matrice de moins en moins dense sans toutefois que des nœuds soient totalement isolés des autres. Finalement, les valeurs que peuvent prendre la proportion d'attaquants et la proportion de paquets qu'ils jettent couvrent tout l'intervalle de valeurs possibles. Il est bon de noter que le cas où tous les nœuds sont attaquants n'est utilisé que lors de la calibration afin de pouvoir vérifier facilement le fonctionnement de l'implémentation de l'attaquant.

Calibration

Ces expériences ont pour but de calibrer le modèle et de valider son fonctionnement à petite échelle. Cette configuration doit permettre de valider le fonctionnement des

différentes fonctions. Elle est très simple et met en place trois nœuds positionnés sur une ligne. La matrice d'adjacence du réseau est donc la suivante :

$$C(t) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, 0 \leq t \leq T$$

Les niveaux de facteurs ont été choisis afin de pouvoir valider sommairement le fonctionnement du modèle. Tout d'abord, l'impact du nombre de paquet est validé. Par la suite, c'est l'impact des attaquants sur le comportement du modèle qui est vérifié. Pour chacune de ces configurations, un seul scénario est construit étant donné que la topologie est fixe et les matrices d'attaques constantes.

TABLEAU 4.2 Scénario 3 nœuds

Nombre de scénarios	Trafic	Proportion Attaquants	Paquets Jetés
	(Paquets/s)		
1	10	0	0
1	100	0	0
1	10	1	0,5
1	10	1	1

Validation

Ces expériences ont pour but de valider le fonctionnement du modèle. C'est lors de cette phase que seront comparés les résultats des indices afin de vérifier si le modèle reflète bien le fonctionnement des RAHM.

Une configuration de base est tout d'abord validée. Cette configuration contient un nombre peu élevé d'attaquants (25%) qui tentent de rester inaperçus en laissant tomber le cinquième des paquets qu'il devrait acheminer. Le terrain est relativement grand (200 m $\times$ 200 m). Ainsi, il est probable que les paquets aient besoin d'être acheminés sur des routes comprenant plus d'un saut. Chacun des nœuds envoie sur le réseau 1 paquets/s vers chacun des autres nœuds. Cela cause des collisions sans toutefois paraliser totalement le réseau. Le nombre de nœuds est quant à lui suffisament élevé pour pouvoir observer le comportement du modèle sans toutefois que son exécution soit trop longue.

TABLEAU 4.3 Scénario de base

Nombre de scénarios	Nombre Noeuds	Trafic	Distance	Proportion Attaquants	Paquets Jetés
	(Noeuds)	(Paquets/s)	(m)		
5	20	1	200	0,25	0,25

Ensuite, une étude de sensibilité sur chacun des facteurs est effectuée. Ainsi, il est possible de voir quels facteurs influent sur la précision du modèle tout en évitant un trop grand nombre de configurations. Tout d'abord, l'impact du nombre de paquets sur la précision du modèle est validé. Le nombre de paquets fait varier la quantité de collisions. Le nombre de paquets y est toujours inférieur à la capacité du canal.

Voici donc les niveaux des facteurs qui permettront de valider le fonctionnement du modèle et l'influence du nombre de paquets sur sa précision.

TABLEAU 4.4 Nombre paquets

Nombre de scénarios	Nombre Noeuds	Trafic	Distance	Proportion Attaquants	Paquets Jetés
	(Noeuds)	(Paquets/s)	(m)		
5	20	0.5	200	0,25	0,25
5	20	5	200	0,25	0,25

Par après, une étude de sensibilité est effectuée sur l'impact de la topologie sur la précision du modèle. La matrice de connectivité qui sera déduite de l'emplacement des nœuds aura un impact sur la création des routes. Il est donc important de quantifier cet impact.

TABLEAU 4.5 Distance

Nombre de scénarios	Nombre Noeuds	Trafic	Distance	Proportion Attaquants	Paquets Jetés
	(Noeuds)	(Paquets/s)	(m)		
5	20	1	1	0,25	0,25
5	20	1	100	0,25	0,25
5	20	1	300	0,25	0,25
5	20	1	400	0,25	0,25

Le tableau 4.6 présente les niveaux qui seront utilisés afin de décrire l'attaquant qui est présent dans le réseau. Une étude de sensibilité est donc effectuée sur la proportion d'attaquants et sur la proportion de paquets qu'ils jettent.

TABLEAU 4.6 Attaquants

Nombre de scénarios	Nombre Noeuds	Trafic	Distance	Proportion Attaquants	Paquets Jetés
	(Noeuds)	(Paquets/s)	(m)		
5	20	1	200	0	0
5	20	1	200	0,5	0,25
5	20	1	200	0,75	0,25
5	20	1	200	0,99	0,25
5	20	1	200	0,25	0,50
5	20	1	200	0,25	0,75
5	20	1	200	0,25	0,99

Finalement, des configurations sont simulées afin de déterminer si le modèle peut être utilisé pour un réseau comprenant un très grand nombre de nœuds, par exemple plus de 1000. Ces tests, dont les configurations sont montrés dans le tableau 4.7, permettent également de vérifier si l'exécution du modèle est plus rapide que celle du logiciel Qualnet.

TABLEAU 4.7 Temps calcul

Nombre de scénarios	Nombre Noeuds	Trafic	Distance	Proportion Attaquants	Paquets Jetés
	(Noeuds)	(Paquets/s)	(m)		
1	2	1	1	0	0
1	10	1	1	0	0
1	20	1	1	0	0
1	30	1	1	0	0
1	40	1	1	0	0

4.6 Exécution des tests et analyse statistique

Après avoir défini le plan d'expérience, il est possible d'effectuer des tests qui seront significatifs et d'obtenir des résultats qui seront reproductibles. Toutefois, le sens que prendront ces résultats doit être défini en fonction d'une analyse statistique rigoureuse. De plus, le temps qu'auront pris les différents tests à s'exécuter n'aura de sens qu'en fonction d'un environnement de test bien défini.

Cette section décrit tout d'abord l'environnement de test. Ensuite, les tests statistiques utilisés sont présentés. Comme tous les graphiques présentés sont de la même forme, celle-ci est décrite dans uen courte sous-section. Finalement, les résultats sont exposés afin d'évaluer la performance du modèle.

4.6.1 Environnement de test

Le temps d'exécution pour un simulateur est toujours un facteur déterminant. En effet, même si le simulateur est très précis, un temps d'exécution trop élevé peut rendre son emploi impossible. Toutefois, afin de pouvoir qualifier le temps d'exécution, il est nécessaire de connaître l'environnement de test. Ce dernier a un impact tout aussi important que le simulateur utilisé sur le temps d'exécution.

Les tests ont été effectués sur un ordinateur Intel Pentium 4 ayant 1,24 Go de mémoire vive et un processeur cadencé à 2,79 GHz. Le système d'exploitation est Windows XP SP2. La version de Qualnet utilisée est 3.9. Le compilateur C++ Microsof .Net est utilisé pour compiler Qualnet tandis que le compilateur Visual Studio 6.0 est utilisé pour compiler l'implantation du modèle. Pour les calculs statistiques, Matlab version 7.0 est utilisé.

4.6.2 Tests statistiques

Chaque scénario défini précédemment a été simulé avec le modèle ainsi qu'avec le logiciel Qualnet. Les résultats obtenus pour chacun de ces scénarios ne sont toutefois pas identiques. Chaque relance étant effectuée de façon pseudo-aléatoire, il est difficile de reproduire une même expérience deux fois, à moins d'utiliser la même graine d'initialisation (*seed*). Pour comparer les résultats venant des deux simulateurs, il est donc nécessaire d'utiliser un test statistique.

Pour cela, il est nécessaire d'utiliser un test d'ajustement comme le test de Kolmo-

gorov-Smirnov. Ce test d'ajustement est utilisé afin de vérifier que les distributions obtenues à l'aide de Qualnet obéissent à la même loi et donc que le modèle décrit bien les RAHM.

Voici les hypothèses qui sont posées pour le test :

$$H_0 : M(x) = Q(x)$$

$$H_1 : M(x) \neq Q(x)$$

Où $M(x)$ correspond à la distribution obtenue à l'aide du modèle, et $Q(x)$ correspond à la distribution obtenue à l'aide de Qualnet.

Le test utilisé est implanté dans Matlab dans la fonction kstest2. Le paramètre utilisé est $\alpha = 0,05$. La probabilité de rejeter l'hypothèse H_0 alors que l'on aurait dû l'accepter est donc égale à $\alpha = 0,05$.

4.6.3 Explication des graphiques

Les graphiques qui sont présentés dans ce chapitre sont tous de la même forme, exepté la figure 4.8 qui montre la progression du temps de calcul en fonction du nombre de nœuds. Les autres graphiques montrent en ordonnée la métrique observée et en abscisse l'indice i de l'émetteur pour qui cette métrique se rapporte. L'indice j du nœud destination est quant à lui représenté sur l'axe qui est perpendiculaire au plan. Pour un nœud i donné, plusieurs nœuds j s'y rapportent ce qui explique pourquoi il y a plusieurs points pour un même i.

4.6.4 Calibration

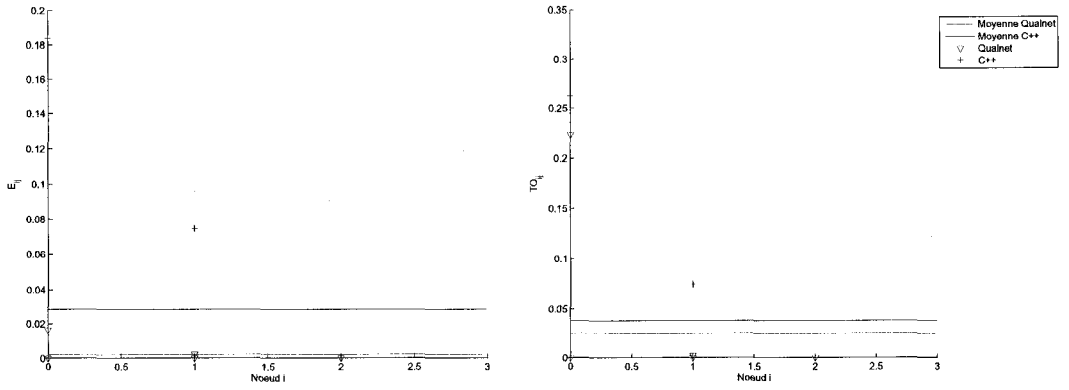
La calibration du modèle est une étape qui est nécessaire afin de valider le fonctionnement général du modèle.

La calibration utilise un scénario comprenant trois nœuds et a pour but de valider l'implémentation du modèle et les différentes fonctions ajoutées à Qualnet. Les figures 4.4(a) à 4.7(b) montrent clairement que les résultats obtenus à l'aide du modèle suivent les mêmes tendances que les résultats obtenus à l'aide de Qualnet.

En effet, on voit tout d'abord que lorsqu'aucun attaquant n'est présent (figures 4.4(a) à 4.5(b)), $TO \approx E$. Ceci appuie l'hypothèse faite à la section 4.2 que $D = 1$. Ensuite, on peut voir que l'augmentation du trafic fait augmenter les pertes de

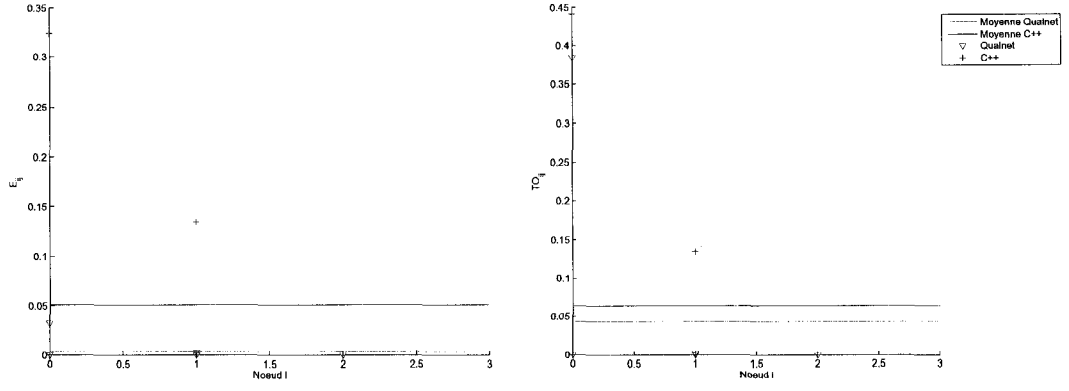
paquets causées par l'environnement (figures 4.5(a) et 4.5(b)). La probabilité $\Pr(E = 1)$ augmente avec la quantité de trafic. Aussi, on peut voir que lorsque le trafic est très peu élevé, $TO \approx J'$ (figures 4.6(a) à 4.7(b)).

On peut toutefois remarquer que bien que les tendances sont respectées, c'est à dire que les différents paramètres influencent de la même façon les métriques venant du modèle que celles venant de Qualnet, les différences entre les deux sont constantes et importantes. Lors de la validation du modèle, on peut voir que le test statistique montre que, sauf exception, les échantillons fournis par le modèle et Qualnet ne proviennent pas de la même distribution de probabilité.



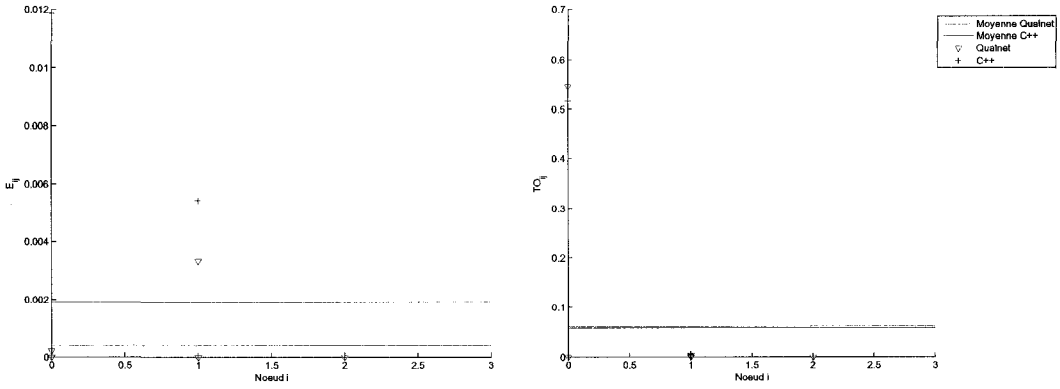
(a) (b)

FIGURE 4.4 Calibration, 10 paquets/s avec (a) $\Pr(E_{ij})$ et (b) $\Pr(TO_{ij})$.



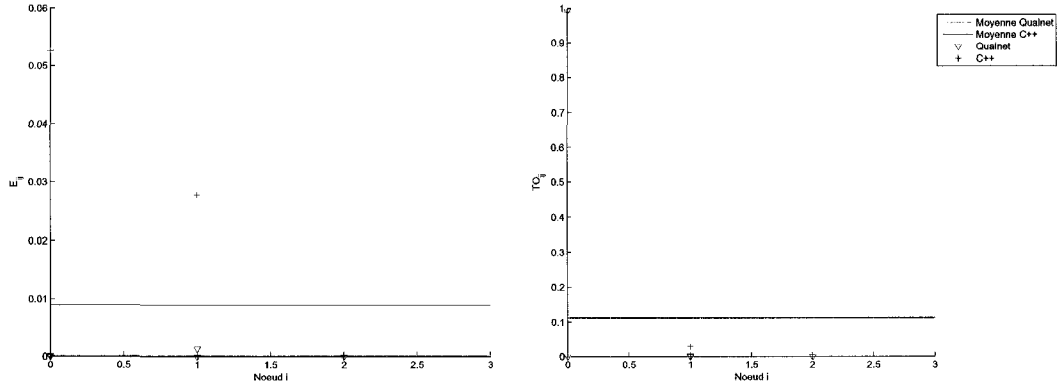
(a) (b)

FIGURE 4.5 Calibration, 100 paquets/s aucun attaquant avec (a) $\Pr(E_{ij})$ et (b) $\Pr(TO_{ij})$.



(a) (b)

FIGURE 4.6 Calibration, 10 paquets/s, 50% d'attaquants avec (a) $\Pr(E_{ij})$ et (b) $\Pr(TO_{ij})$.



(a) (b)

FIGURE 4.7 Calibration, 10 paquets/s, 99% d'attaquants avec (a) $\Pr(E_{ij})$ et (b) $\Pr(TO_{ij})$.

4.6.5 Validation

Le plan d'expérience a été réalisé dans l'environnement de test défini. Les résultats bruts ont été analysés à l'aide du logiciel Matlab afin d'obtenir les différents résultats statistiques et les graphiques montrant les indices de performance mesurés. Tout d'abord, la progression du temps de calcul est montrée. Ensuite, l'analyse statistique permettant de valider le modèle est effectuée. Les résultats de ces analyses sont par la suite commentés dans une autre section.

Temps de calcul. À l'aide de la figure 4.8, il est possible de voir que le temps de calcul de Qualnet augmente beaucoup plus rapidement que celui du modèle. Les calculs effectués par Qualnet sont donc beaucoup plus long que ceux effectués par le modèle et ont une complexité algorithmique qui semble plus importante. C'est un avantage marqué pour le modèle.

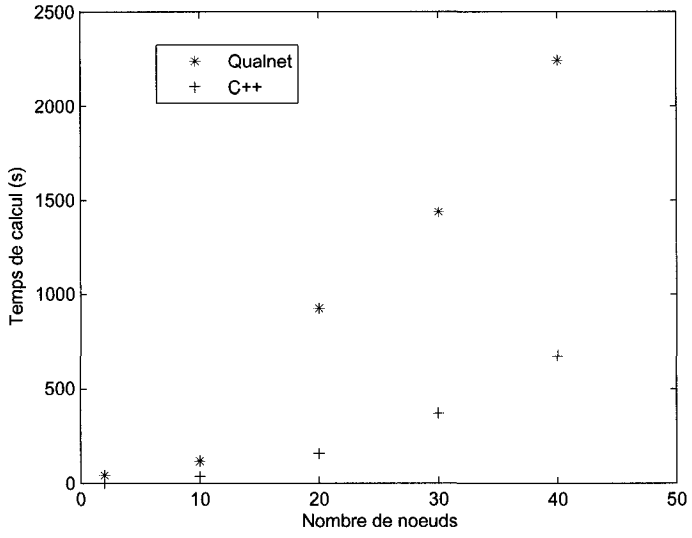


FIGURE 4.8 Temps de calcul

Analyse statistique. Les résultats obtenus par Qualnet et le modèle ne proviennent clairement pas de la même distribution de probabilité. Le test de Kolmogorov-Smirnov montre que le modèle ne décrit pas adéquatement le fonctionnement des RAHM.

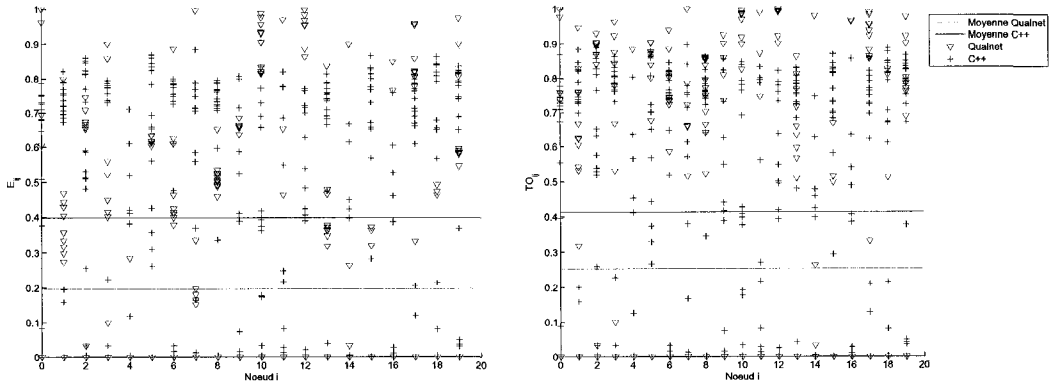
En effet, pour $\Pr(E_{ij}(t) = 1)$, $\Pr(T_{ij}(t) = 1)$, il faut rejeter l'hypothèse H_0, c'est-à-dire l'hypothèse voulant que $M(x) = Q(x)$ où $M(x)$ correspond à la

distribution du modèle et $Q(x)$ correspond à la distribution des résultats obtenus à l'aide de Qualnet.

Les résultats sont visibles sur les figures 4.9(a) à 4.13(b) où il est possible de voir que les résultats venant de Qualnet sont complètement différents de ceux venant du modèle. Il est bon de noter que le modèle ne permet pas de déterminer une borne supérieure ou inférieure pour les différentes métriques. En effet, pour les figures 4.9(a) et 4.9(b), le modèle surestime les valeurs des métriques tandis que pour les figures 4.10(a) et 4.10(b), il les sous-estime.

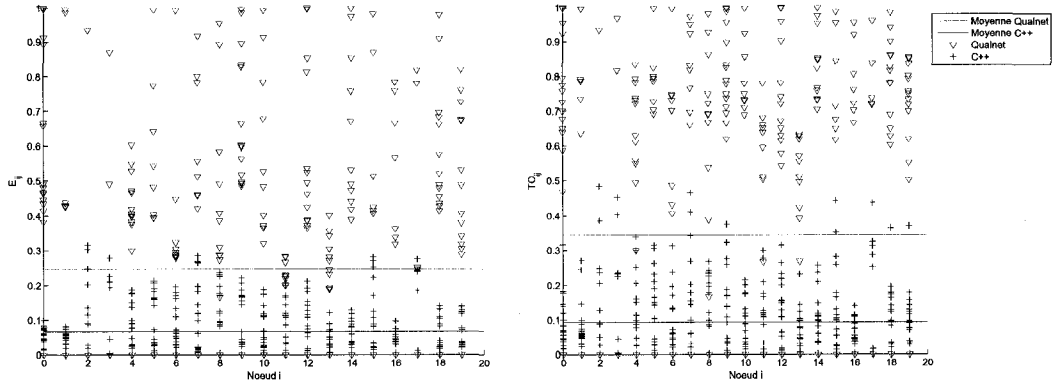
4.7 Synthèse des résultats

Dans cette section, les résultats venant des tests statistiques sont analysés afin de voir si le modèle est adéquat et si des résultats généraux peuvent être tirés de cette expérience.



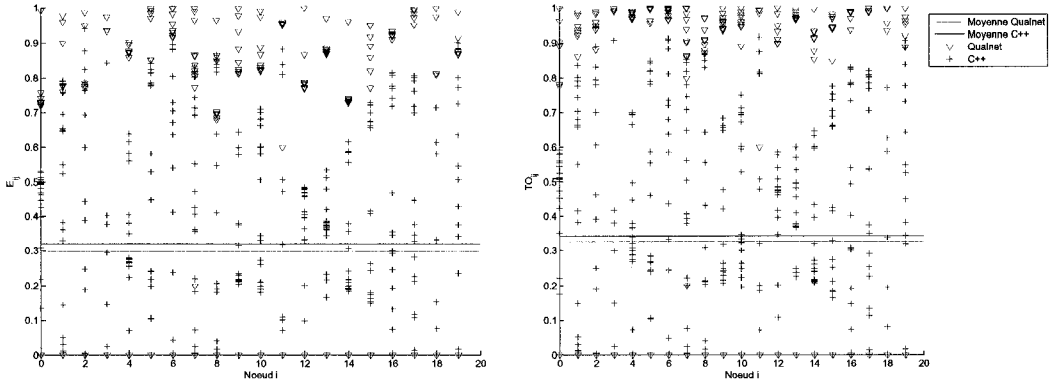
(a) (b)

FIGURE 4.9 Cas de base, avec (a) $\Pr(E_{ij})$ et (b) $\Pr(TO_{ij}(t))$.



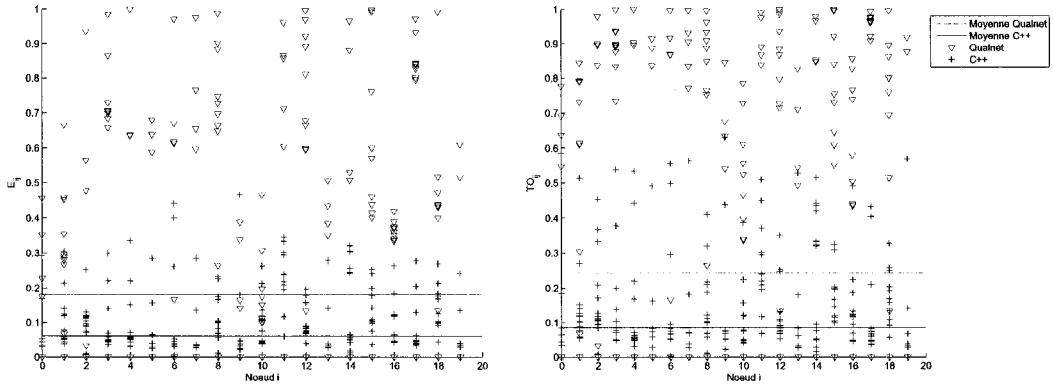
(a) (b)

FIGURE 4.10 Impact du trafic à 0,5 paquet/s sur $\Pr(E_{ij})$, (a) et $\Pr(TO_{ij})$, (b).



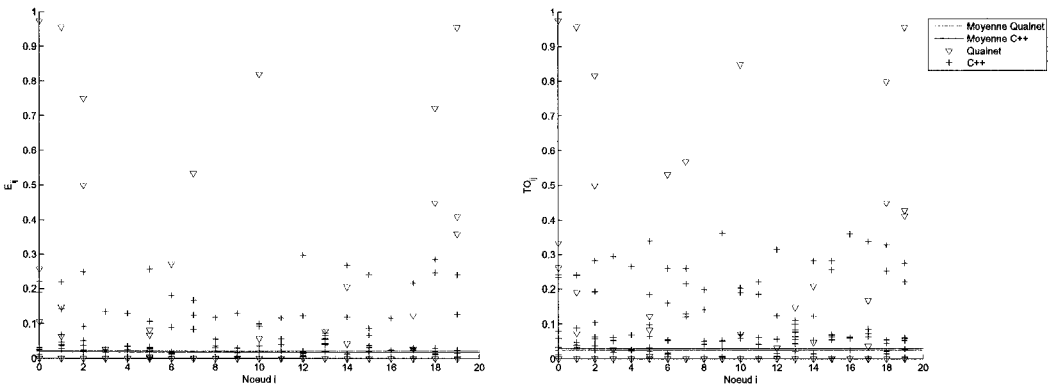
(a) (b)

FIGURE 4.11 Impact du trafic à 5 paquets/s, sur $\Pr(E_{ij})$, (a), et $\Pr(TO_{ij})$, (b).



(a) (b)

FIGURE 4.12 Impact du taux d'attaquants 99% d'attaquants, sur $\Pr(E_{ij})$, (a), et $\Pr(TO_{ij})$, (b).



(a) (b)

FIGURE 4.13 Impact de la distance, terrain carré de $0,16km^2$, sur $\Pr(E_{ij})$, (a), et $\Pr(TO_{ij})$, (b).

Les tests statistiques montrent que les probabilités obtenues à l'aide de Qualnet et du modèle sont différentes,ce qui est facilement observable sur les différentes figures. Ceci peut être dû à plusieurs facteurs. Un des facteurs est que la modélisation des pertes de paquets ne tient pas compte de toutes les caractéristiques du protocole 802.11. En effet, les messages d'erreur ne sont pas pris en compte. Une autre cause possible de cette grande disparité est que les résultats du moment précédent sont utilisés afin de résoudre les systèmes d'équations non-linéaires tel que décrit à la section 4.3. Cette hypothèse n'a toutefois pas été testée.

Mais le principal facteur d'erreur est que l'hypothèse voulant que le taux de trafic causant les différentes interférences soit de distribution exponentielle semble fausse. Étant donné que les paquets prennent un temps minimum pour être transmis, c'est-à-dire $T_{\text{DIFS}} + T_{\text{DATA}}$, la distribution du temps inter-arrivée des paquets émis ne peut pas être exponentielle. De plus, lorsqu'un nœud émet à la pleine capacité du canal, la distribution du temps inter-arrivée sera constante et égale au délai minimum.

Une incohérence amenée par cette hypothèse est qu'un nœud aurait dû pouvoir émettre un paquet de données et un accusé réception simultanément pour que l'hypothèse soit vraie, ce qui est bien sûr impossible. En effet, pour que le temps entre l'émission de deux paquets consécutifs soit exponentiel, il ne peut pas tenir compte du fait qu'un nœud doit aussi répondre aux paquets que les autres nœuds lui envoient en émettant un accusé réception. Le calcul du trafic utilisé dans les équations 3.17 à 3.20 est donc faussé étant donné qu'il ne tient pas compte de cette contrainte physique. Toutefois, en analysant l'équation 3.24, on peut voir que cette erreur sera négligeable si $T_{\text{DATA}} \gg T_{\text{ACK}}$ et que ${\Lambda^{\text{b}}}_i \approx {\Lambda^{\text{a}}}_i \cup {\Lambda^{\text{b}}}_i \approx {\Lambda^{\text{c}}}_i \cup {\Lambda^{\text{b}}}_i \approx {\Lambda^{\text{d}}}_i$. Sinon, il faudra tenir compte de ce phénomène. Dans le cas des tests qui ont été effectués, $T_{\text{DATA}} \gg T_{\text{ACK}}$. Comme le trafic suit une distribution uniforme, il est aussi probable que les deux conditions soient rencontrées et que l'impact de cette incohérence soit minime.

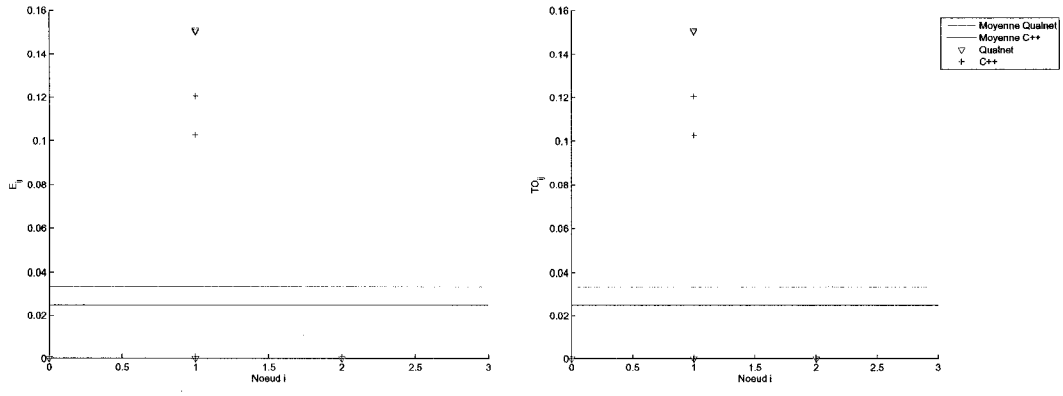
Le temps inter-arrivée des paquets reçus par les nœuds à proximité de l'émetteur ne pourra donc en aucun cas être exponentiel. Comme une partie de ces paquets devront être réacheminés par les nœuds, même le temps inter-arrivé des paquets devant être émis ne sera pas exponentiel.

L'hypothèse en question sera valide seulement dans le cas où très peu de paquets sont émis et que le nombre de collisions est très faible. Ainsi, le temps inter-arrivé des paquets venant de la couche application et venant des autres nœuds sera largement supérieur au temps d'émission des paquets et on pourra alors négliger ce dernier.

Comme il n'y aura que très peu de collisions, un paquet généré par la couche application sera immédiatement émis et retransmis par les nœuds intermédiaires, ce qui fait que le temps d'attente avant d'être émis sera approximativement égal à zéro. Si la quantité de paquet est élevée, cette hypothèse est toutefois déraisonnable. Lors de travaux futurs, il faudra donc modéliser autrement la probabilité de perdre un paquet à cause des interférences.

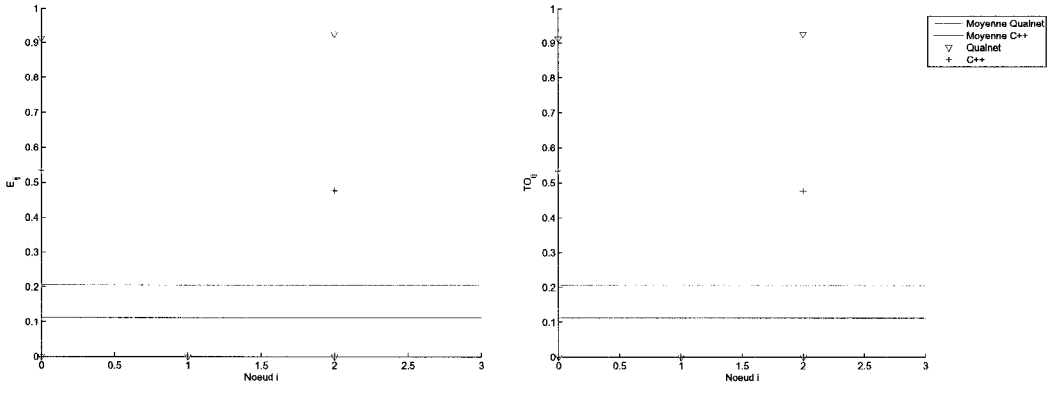
Un autre résultat important est qu'il est impossible de déduire des métriques à l'aide des valeurs moyennes des différents facteurs. Au chapitre précédent, à la section 3.8, ce résultat avait été obtenu à l'aide du modèle. Les figures 4.14(a) à 4.14(a) montrent cette caractéristique. En effet, dans le scénario ayant généré les figures 4.14(a) à 4.14(b), un nœud central envoie des paquets à deux autres nœuds qui lui sont voisins mais qui ne peuvent communiquer directement entre eux. Dans le scénario ayant généré les figures 4.15(a) à 4.15(b), les deux nœuds envoient au nœud central des paquets. Dans le premier cas, il y a très peu de collisions tandis que dans le second, le réseau est surchargé. En inversant simplement les flots, et donc en gardant les flots moyens égaux, le taux de pertes de paquets a explosé. Dans le premier cas, les nœuds pouvaient sans problème recevoir les paquets du nœud central tandis que dans le second, cela est impossible. On voit donc clairement que bien que les caractéristiques moyennes du réseaux soient les mêmes, il est impossible de tirer une conclusion et ce même avec trois nœuds. Il faut donc résoudre complètement le modèle afin d'obtenir des résultats valides. Ce résultat est valide étant donné qu'il a été observé tant à l'aide du modèle qu'à l'aide de Qualnet.

Bien que les résultats montrent que le modèle n'est pas adéquat, deux résultats importants ont pu être trouvés. Premièrement, on voit que les délais inter-arrivés des paquets devant être émis et des paquets émis par un nœud ne sont pas exponentiels. Ensuite, on peut constater qu'il est impossible de tirer des conclusions à l'aide des valeurs moyennes, chose que plusieurs auteurs tentent.



(a) (b)

FIGURE 4.14 Cas particulier, peu de collisions avec (a) $\Pr(E_{ij})$ et (b) $\Pr(TO_{ij})$.



(a) (b)

FIGURE 4.15 Cas particulier, beaucoup de collisions avec (a) $\Pr(E_{ij})$ et (b) $\Pr(TO_{ij})$.

CHAPITRE 5

Conclusion

Ce dernier chapitre a pour but de résumer l'ensemble du mémoire et de présenter les différentes caractéristiques du modèle. Tout d'abord, la synthèse des travaux qui ont été présentés dans ce mémoire et leur originalité est présentée. Ensuite, leurs limites sont exposées. Finalement, les travaux qui devront être effectués afin d'améliorer le modèle sont exposés.

5.1 Synthèse des travaux et originalité des contributions

L'objectif principal de ce mémoire était de créer un modèle analytique décrivant la relation entre l'environnement des RAHM, les attaquants et des indices de performances définis. Cet objectif principal était scindé en trois objectifs secondaires, c'est-à-dire :

- Analyser différents protocoles de routage sécuritaire utilisant le concept de réputation et définir les différentes attaques visant les RAHM.
- Définir un modèle décrivant la relation entre l'environnement des RAHM, les attaquants agissant dans les RAHM, le taux de faux positifs, le taux de faux négatifs et le taux de perte de paquets.
- Valider le modèle en le comparant avec un simulateur.

Dans cette section, ces objectifs seront exposés et l'atteinte de ceux-ci sera discutée.

5.1.1 Analyse des CMUR existantes

Cet objectif avait pour but d'analyser différents protocoles de routage sécuritaire utilisant le concept de réputation et définir les différentes attaques visant les RAHM.

Avant d'analyser les CMUR, il était important de définir les RAHM dans leur

ensemble. La revue de littérature a donc présenté leur principales caractéristiques en insistant sur les attaques auxquelles ces réseaux font face.

Des mécanismes préventifs ont ensuite été exposés. Ceux-ci ont pour but principal que les tables de routages se créent de façon conforme au protocole. Tout d'abord, des méthodes de distribution des clefs et d'authentification ont été décrites. Ensuite, différents protocoles utilisant ces méthodes ont été introduits, tout comme différentes CMUR. Celles-ci tentent d'identifier et d'exclure les nœuds malicieux du réseau afin de permettre son bon fonctionnement. L'analyse de performance que les auteurs font de leurs protocoles tendent à démontrer que ceux-ci sont très efficaces même si la quantité d'attaquant est très élevée.

Par contre, il a été possible de voir que ces analyses sont souvent inadéquates. Les critères de performances utilisés sont parfois déficients, les attaquants qui y sont présents sont souvent peu coordonnés et ne tentent pas de passer inaperçus, ce qui rend leur détection très facile. De plus, pour valider les analyses de performance, il est nécessaire de vérifier le code qui est modifié au sein des simulateurs, tout comme les différents fichiers de configuration.

Il serait donc utile de pouvoir utiliser un modèle analytique afin de pouvoir procéder à cette validation. Toutefois il n'existait aucun autre modèle que celui qui est proposé qui puisse déterminer quelle sera l'efficacité d'une CMUR dans un environnement donné avec un attaquant défini. L'objectif initial du modèle proposé est donc de calculer de façon analytique les taux de faux positifs de faux négatifs et de pertes de paquets pour les nœuds honnêtes ou malicieux en tenant compte de l'environnement et des attaquants et de valider les analyses de performances faites à l'aide de simulateurs.

5.1.2 Définition du modèle

Cet objectif visait à définir un modèle décrivant la relation entre la demande, l'environnement des RAHM, les attaquants agissant dans les RAHM, le taux de faux positifs, le taux de faux négatifs et le taux de perte de paquets. Un modèle basé sur les processus stochastique a donc été présenté. Un tel modèle permettrait de faire une analyse de performance qui ne dépendrait pas du simulateur utilisé et faciliterait la validation des CMUR qui sont proposées. De plus, il ne serait plus nécessaire de valider l'ensemble du code ayant servi aux simulations.

La topologie du RAHM y est modélisé par une matrice d'adjacence qui est elle-même une matrice de processus stochastiques. Cette matrice détermine quels nœuds sont en mesure de s'entendre entre eux à un temps t. Les CMUR qui sont en place sont caractérisées par un ensemble de processus stochastiques qui déterminent la probabilité qu'un nœud achemine les paquets des autres nœuds, tout comme les probabilités que les paquets soient jetés.

L'attaquant qui agit au sein du réseau peut tout aussi bien mener des attaques coordonnées que de tenter d'influencer le calcul des réputations en diffamant ou en augmentant la réputation des nœuds qui composent tout comme lui le RAHM.

Les indices de performance pertinents qui ont été identifiés lors de la revue de littérature sont aussi modélisés par des processus stochastiques. Ces indices sont le taux de faux positifs, le taux de faux négatifs, le taux de pertes de paquets des nœuds honnêtes et le taux de pertes de paquets des nœuds malicieux.

Aucun autre modèle analytique ne permet d'obtenir des valeurs pour ces indices de performance, les modèles existants se limitant en général à l'analyse du taux de pertes de paquets par des files d'attentes ou à tenter de prévoir l'efficacité des CMUR en utilisant la théorie des jeux. Le modèle proposé permet quant à lui d'analyser des CMUR tout en tenant compte de l'influence de l'environnement.

Il est toutefois bon de souligner que le modèle et les résultats expérimentaux montrent que même pour des scénarios très simples, il est impossible de trouver une relation entre les différentes métriques et les valeurs moyennes des paramètres de configuration.

5.1.3 Validation du modèle

Le dernier objectif était de valider le modèle en vérifiant l'adéquation des prédictions de performance établies par ce modèle avec les résultats obtenus à l'aide d'un simulateur.

Le modèle a donc été implanté à l'aide du language C++ afin de simuler un RAHM où aucune réputation n'est utilisée et sans mobilité. Les résultats du modèle ont été comparés à ceux obtenus à l'aide de Qualnet, un simulateur commercial.

Tout d'abord, seuls certains aspects du modèle ont été simulés. Ces aspects sont primordiaux dans le calcul des réputations. Il sera donc nécessaire, lors de travaux futurs, d'implanter des CMUR telles que celles présentées au chapitre 2. Ensuite, il a

été démontré statistiquement que les résultats sont différents de ceux obtenus à l'aide du simulateur Qualnet. Le modèle, tel que présenté, ne peut donc pas être utilisé à lui seul pour caractériser les CMUR car il n'est pas suffisamment précis. Il est donc encore nécessaire d'utiliser un simulateur reconnu.

Toutefois, des résultats complémentaires très importants ont été obtenus. Ainsi, il a été possible de déterminer que la distribution du temps inter-arrivée des paquets reçus par les nœuds ne suit pas une distribution exponentielle, ce qui invalide une hypothèse de base. Ceci est la principale cause du mauvais calcul de l'environnement. Ce résultat aidera, lors de travaux futurs, à rendre le modèle adéquat.

Aussi, il a été montré que la moindre perturbation peut changer radicalement les résultats et que se baser sur des valeurs moyennes afin de trouver la valeur moyenne des pertes de paquets est impossible. Ainsi, il est impossible de caractériser une CMUR globalement sans tenir compte de l'environnement et de la demande propre à un scénario. Ceci a été observé à l'aide du simulateur Qualnet et est donc indépendant de la validité du modèle.

5.2 Limites des travaux

Bien que le modèle réussisse à décrire la tendance des différentes métriques caractérisant les RAHM, le modèle et sa validation comportent plusieurs limites. Deux d'entres elles sont particulièrement importantes et concernent la modélisation déficiente de l'environnement et la simulation qui ne couvre que certains aspects du modèle. Voici une description plus approfondie de ces limites.

Modélisation de l'environnement. La plus grande limite du modèle est qu'il ne modélise pas adéquatement l'environnement. Il faudra donc revoir certaines hypothèses afin d'en améliorer la précision.

Simulations limitées. Une autre grande limite de l'analyse de performance est qu'en plus de ne pas avoir considéré la mobilité dans les scénarios, un seul protocole de routage a été analysé et ce, sans CMUR. Il aurait donc été impossible de prouver que le modèle est adéquat avec ce seul mémoire. L'objectif de l'analyse de performance n'étant que de valider certains aspects du modèle, il sera nécessaire, lors de travaux futurs, de la compléter en modélisant d'autres protocoles de routages et d'autres CMUR.

5.3 Travaux futurs et leçons apprises

Bien que le modèle ait été décrit dans son ensemble au chapitre 3, l'analyse de performance n'a porté que sur certains aspects critiques. Ainsi, la mobilité est exclue et seules les métriques sur lesquelles se basent le calcul des réputations ont été validées. Il y a donc différents aspects du modèle qui doivent être améliorés. De plus, le calcul de la probabilité $\Pr(E_{ij}(t) = 1)$ devra lui aussi être amélioré. Voici donc un résumé des travaux qui devront être effectués ultérieurement.

Calcul des pertes dues à l'environnement. Avant d'implanter des CMUR dans le modèle, il est nécessaire d'améliorer le calcul de cette probabilité. En effet, l'hypothèse voulant que le délai entre l'émission de deux trames consécutives suive une distribution exponentielle est inadéquate. Il faudra donc revoir la modélisation de ce calcul.

Implémentation des CMUR. Il sera nécessaire d'implanter des CMUR afin de voir dans quel cas le calcul des réputations est adéquat et de valider le modèle plus à fond. En effet, bien que le calcul des métriques utilisées par les CMUR a été validé, il faudra valider l'interaction entre les CMUR et l'environnement. En effet, les flots seront affectés par les CMUR ce qui pourrait influencer le calcul des autres métriques, y compris des CMUR elles-mêmes.

Mobilité. L'implantation de la mobilité est nécessaire afin de terminer la validation complète du modèle. L'implantation en C++ devra donc être modifiée. Il sera nécessaire d'améliorer la classe *cTopologie* afin de permettre la prise en charge de la mobilité.

En plus des deux résultats complémentaires qui ont pu être apportés lors des simulations, il est possible de tirer des enseignements de ce mémoire. Le principal est que la modélisation complète d'un RAHM est beaucoup trop complexe pour être résolue analytiquement et rend même l'emploi des méthodes numériques très difficile. En effet, simplement trouver les probabilités de pertes de paquets dues à l'environnement nécessite la résolution d'un système d'équations non-linéaires comprenant des processus stochastiques qui ne sont pas exponentiels et donc avec mémoire.

De plus, comme tout modèle devra être implémenté et que cette implémentation sera complexe et dépendante du protocole utilisé, il sera nécessaire de valider l'implémentation du modèle afin de valider une CMUR dans un environnement précis. Par conséquent, l'intérêt que l'on peut porter vers un tel modèle s'amenuise. Il serait donc

peut-être plus efficace de bâtir un ensemble de tests standardisé qui pourrait servir à valider les CMUR.

Références

ANDEREGG, L. et EIDENBENZ, S. (2003). Ad hoc-VCG : A Truthful and Cost-Efficient Routing Protocol for Mobile Ad hoc Networks with Selfish Agents. *Proceedings of International Conference on Mobile Computing and Networking, MobiCom.* 245–259.

BUCHENNER, S. et LE BOUDEC, J. Y. (2002a). Nodes bearing grudges : towards routing security, fairness and robustness in mobile ad hoc networks. *Proceedings of Euromicro Workshop on Parallel, Distributed and Network-based Processing, EUROMICRO-PDP.* 403–410.

BUCHENNER, S. et LE BOUDEC, J. Y. (2002b). Performance Analysis of the CONFIDANT Protocol. *Proceedings of Mobile Ad Hoc Networking and Computing, MobiHoc.* 226–236.

BUTTYÀN, L. et HUBAUX, J. P. (2001). Nuglets : a virtual currency to stimulate cooperation in self-organized mobile ad hoc network. *Technical Report, DSC-2001-001, Institute for Computer Communications and Applications, EPFL*, 1–15.

CAPKUN, S., HUBAUX, J. et BUTTYÀN, L. (2003). Mobility Helps Security in Ad Hoc Networks. *Proceedings of Mobile Ad Hoc Networking and Computing, MobiHoc.* 46–56.

CARDENAS, A., BENAMMAR, N., PAPAGEORGIOU, G. et BARAS, J. S. (2004). Cross-layered security analysis of wireless ad hoc networks. *Proceedings of Army Science Conference, ASC.* 1–2.

CORSON, S. et MACKER, J. (1999). *RFC 2501 : Mobile Ad hoc Networking (MANET) Routing Protocol Performance Issues and Evaluation Considerations.* IETF. Status : INFORMATIONAL.

FARIA, D. (2006). Modeling Signal Attenuation in IEEE 802.11 Wireless LANs - Vol. 1. *Technical Report, TR-KP06-0118, Kiwi Project, Stanford University*, 1–5.

HASSWA, A., ZULKERNINE, M. et HASSANEIN, H. (2005). Routeguard : An Intrusion Detection and Response Systm for Mobile Ad Hoc Networks. *Proceedings*

of IEEE International Conference on Wireless and Mobile Computing, WiMob. 336–343.

HU, Y. C., JOHNSON, D. B. et PERRIG, A. (2002a). SEAD : Secure Efficient Distance Vector Routing for mobile Wireless networks. *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, WMCSA.* 3–13.

HU, Y. C., PERRIG, A. et JOHNSON, D. B. (2002b). Ariadne : a secure on-demand routing protocol for ad hoc networks. *Proceedings of International Conference on Mobile Computing and Networking, MobiCom.* 12–23.

HU, Y. C., PERRIG, A. et JOHNSON, D. B. (2003a). Packet Leashes : a defense against wormhole attacks in Wireless Networks. *Proceedings of Joint Conference of the IEEE Computer and Communications Societies, INFOCOM.* 1976–1986.

HU, Y. C., PERRIG, A. et JOHNSON, D. B. (2003b). Rushing attacks and defense in wireless ad hoc network routing protocols. *Proceedings of Workshop on Wireless Security, WiSe.* 30–40.

JOHNSON, D., MALTZ, D. et HU, Y. (2004). *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR).* IETF. Status : Internet DRAFT.

LIU, J. et ISSARNY, V. (2004). Enhanced Reputation Mechanism for Mobile Ad Hoc Networks. *Proceedings of Trust Management International Conference, iTrust.* 48–62.

MARTI, S., GIULI, T., LAI, K. et BAKER, M. (2000). Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. *Proceedings of International Conference on Mobile Computing and Networking, MobiCom.* 255–265.

MICHIARDI, P. et MOLVA, R. (2002). CORE : a collaborative reputation mechanism to enforce node cooperation in MANET. *Proceedings of Communication and Multimedia Security Conference, CMS.* 1–18.

MICHIARDI, P. et MOLVA, R. (2003). A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad hoc Networks. *Proceedings of Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, WiOpt.* 1–4.

MOLVA, R. et MICHIARDI, P. (2003). Security in Ad hoc Networks. *Proceedings of Personal Wireless Communication, PWC.* 56–75.

MONTRESOR, A., CARO, G. D. et HEEGAARD, P. E. (2003). Architecture of the Simulation Environment. *Technical Report, IST-2001-38923, Biology-Inspired techniques for Self Organization in dynamic Networks, Università di Bologna*, 1–39.

NURMI, P. (2004). Modelling Routing in Wireless Ad Hoc Networks with Dynamic Bayesian Games. *Proceedings of IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, SECON.* 63–70.

ÖZDEMIR, M. et MCDONALD, A. B. (2005). A Queuing Theoric Model of Ad Hoc Wireless LANs. *Proceedings of IEEE International Conference on Wireless and Mobile Computing, WiMob.* 131–137.

PAPADIMITRATOS, P. et HASS, Z. J. (2003). Secure Link State Routing for Mobile Ad Hoc Networks. *Proceedings of Symposium on Applications and the Internet Workshops, SAINT.* 379–383.

PERRIG, A., CANETTI, R., TYGAR, J. et SONG, D. (2002). The TESLA Broadcast Authentification Protocol. *CryptoBytes*, 2–13.

PERRIG, A. et HU, Y. C. (2004). A Survey of Secure Wireless Ad Hoc Routing. *IEEE Security and Privacy*, 28–39.

SANZGIRI, K., DAHILL, B., NEIL, B., SHIELDS, L. C. et BELDING-ROYER, E. M. (2002). A Secure Routing Protocol for Ad Hoc Networks. *Proceedings of IEEE International Conference on Network Protocolsi, ICNP.* 78–87.

STUEDI, P., CHINELLATO, O. et ALONSO, G. (2005). Connectivity in the presence of Shadowing in 802.11 Ad Hoc Networks. *Proceedings of IEEE Wireless Communications and Networking Conference, WCNC.* 2225–2230.

TSENG, Y. et JIANG, J. (2003). Secure bootstrapping and routing in an ipv6-based ad hoc network. *Proceedings of IEEE International Conference on Parallel Processing Workshop, ICPP.* 375–382.

YANG, H., LUO, H., YE, F., LU, S. et ZHANG, L. (2004). Security in mobile ad hoc networks : challenges and solutions. *IEEE Wireless Communications*, 38–47.

ZAPATA, M. G. et ASOKAN, N. (2002). Securing Ad hoc Routing Protocols. *Proceedings of Workshop on Wireless Security, WiSe.* 1–10.

ZHOU, L. et HAAS, Z. J. (1999). Securing ad hoc networks. *IEEE Network*, 24–30.