Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Supervision des Réseaux et Services Ad-Hoc

THÈSE

présentée et soutenue publiquement le 11 décembre 2006

pour l'obtention du

Doctorat de l'Université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Rémi Badonnel

Composition du jury

Rapporteurs :	Serge Fdida	Professeur à l'Université Pierre et Marie Curie - Paris VI
	Philippe Jacquet	Directeur de Recherche à l'INRIA Rocquencourt
Examinateurs :	Alexander Keller	*Research Manager* à IBM Research - Yorktown, NY, États-Unis
	Karl Tombre	Professeur à l'Ecole des Mines de Nancy
	André Schaff	Professeur à l'Ecole Supérieure d'Informatique et Applications de Lorraine
	Radu State	Chargé de Recherche à l'INRIA Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

Je tiens à remercier en premier lieu les rapporteurs, Serge Fdida et Philippe Jacquet, ainsi que l'ensemble des examinateurs de cette thèse pour l'intérêt qu'ils portent à mes travaux de recherche et pour avoir accepté de faire partie de mon jury de thèse.

Je remercie très sincèrement André Schaff, mon directeur de thèse, pour toute l'attention qu'il m'a apportée. Son expérience et ses grandes compétences ont contribué au bon déroulement de ce travail. Qu'il trouve ici les marques de ma reconnaissance et de mon profond respect.

Un très grand merci à mon encadrant de thèse, Radu State, pour la qualité de sa collaboration, ses précieux conseils, son aide constante et pour la façon efficace et amicale avec laquelle il a suivi ce travail. Pendant les nombreuses heures passées ensemble, j'ai beaucoup appris à son contact.

Pour m'avoir fait confiance en m'accueillant au sein de son équipe, j'adresse mes plus chaleureux remerciements à Olivier Festor. J'ai pris beaucoup de plaisir à travailler avec lui. Son dynamisme, ses conseils de tout ordre et sa passion communicative pour la recherche furent très appréciables tout au long de ce travail de thèse.

J'exprime également ma sincère reconnaissance à Alexander Keller et Joe Hellerstein pour m'avoir ouvert les portes de leur équipe de recherche à IBM Research - New York, et pour m'avoir ainsi permis de diversifier mes compétences scientifiques et techniques au sein de leur laboratoire.

Je remercie profondément Robert Cailliau, ingénieur-chercheur au CERN à Genève et co-inventeur du World Wide Web, pour m'avoir fait découvrir le monde de la recherche, il y a quelques années, dans le cadre d'un projet à ESIAL sous la supervision et les conseils avisés de Jacques Guyard.

Mes remerciements vont également à tous les membres de l'équipe MADYNES ainsi qu'à l'ensemble de mes collègues et amis du LORIA - INRIA Lorraine à qui j'exprime ma profonde sympathie.

Je souhaite enfin remercier du fond du cœur ma famille et mes proches pour leur soutien et leurs encouragements tout au long de cette thèse.

A toutes et à tous, merci.

Table des matières

Table des figures

Introduction

Chapitre 1

Introduction générale

1.1 Contexte scientifique et technique

Les réseaux mobiles sans fil ont connu un très fort développement ces dernières années pour répondre à la hausse constante des besoins en mobilité. Les exemples les plus significatifs sont probablement l'engouement pour la téléphonie cellulaire, mais aussi le large déploiement des réseaux locaux sans fil avec la multiplication des points d'accès dans les lieux publics et chez les particuliers. Cette croissance est maintenue par l'augmentation des débits offerts aux utilisateurs avec le développement de la téléphonie de troisième génération [181] et des premières offres commerciales de réseaux sans fil WiMax [152]. L'arrivée de nouvelles normes de transmission de quatrième génération [95] permettra la réelle convergence des différentes technologies mobiles réunies dans un cœur de réseau entièrement sous protocole IP. Ces normes constitueront le socle d'environnements pervasifs dans lesquels les utilisateurs pourront accéder à des services, communiquer, travailler avec les autres usagers en tout lieu, à tout instant et depuis n'importe quel équipement mobile.

Les réseaux ad-hoc représentent une composante clé de cette évolution et leurs fondements seront inévitablement intégrés aux générations futures de réseaux sans-fil. Ces réseaux auto-organisés [162] sont formés spontanément à partir d'un ensemble d'entités mobiles communicantes, sans nécessiter d'infrastructure fixe préexistante telle qu'une station de base ou un point d'accès par exemple. Les entités mobiles constituent en elles-mêmes le réseau. Elles peuvent être de formes variées : ordinateurs portables, téléphones mobiles, assistants électroniques, capteurs et présentent par conséquent des capacités non homogènes en termes de communication, de puissance de calcul et de stockage. Elles sont libres de se déplacer de manière aléatoire et de s'organiser arbitrairement, si bien que la topologie du réseau est fortement dynamique dans le temps et dans l'espace. Les entités mobiles peuvent intervenir en tant que routeurs pour assurer l'acheminement des paquets entre elles par sauts successifs. Elles communiquent donc soit directement lorsqu'elles se trouvent dans le même voisinage direct, soit par communication multi-sauts le cas échéant en faisant appel à des nœuds intermédiaires. Grâce aux réseaux ad-hoc, l'utilisateur peut ainsi déployer son propre réseau très facilement et sans coût supplémentaire. Mais l'apparente simplicité du concept cache de nombreux défis scientifiques et techniques.

La première forme de réseaux sans infrastructure remonte aux années 1970 avec le projet DARPA PRNET (*Packet Radio Network*) [1]. L'objectif consistait à mettre au point un système de communication multi-sauts sans fil dans le cadre d'applications militaires. Le système était capable de s'auto-organiser sans le support d'une infrastructure fixe, à travers la détection de la connectivité radio et l'établissement de stratégies de routage. Une extension de ces travaux

de recherche ont abouti au projet DARPA SURAN (*Survival Radio Network*) [1] qui visait à expérimenter des équipements de taille plus réduite et à définir des protocoles robustes offrant une meilleure tolérance aux fautes ainsi qu'un meilleur passage à l'échelle.

La commercialisation des premiers réseaux ad-hoc s'est faite discrètement à travers le développement du *Bluetooth* [141]. Initialement proposée par Ericsson en 1994, puis reprise dans le cadre du groupe d'intérêt *Bluetooth SIG* regroupant différents industriels majeurs tels qu'IBM, Microsoft et Motorola, cette technologie permet de transmettre des données entre des équipements périphériques sur une faible distance avec une moindre consommation électrique. Elle peut ainsi former des réseaux particuliers appelés *scatternet* qui requièrent l'utilisation de protocoles de routage multi-sauts. La nécessité de développer des standards ouverts a conduit à la création du groupe de travail MANET (*Mobile Ad-Hoc Networks*) [132] de l'IETF (*Internet Engineering Task Force*) [3] en 1998. Ce groupe est chargé de standardiser les protocoles de routage IP unicast pour les réseaux ad-hoc. La lenteur du processus de normalisation a abouti à une standardisation relativement tardive des protocoles de routage.

Grâce à leur facilité de déploiement et leur faible coût de maintenance, les réseaux ad-hoc sont particulièrement appropriés aux applications militaires telles que les missions d'exploration en terrain difficile et les opérations de secours en situation d'urgence. Cependant, ils présentent de nombreux autres domaines d'applications tels que les communications intervéhiculaires et les réseaux de capteurs. Les possibilités sont décuplées grâce aux développements des réseaux hybrides et des réseaux maillés sans fil. Ainsi, l'interconnexion d'un réseau ad-hoc à une infrastructure fixe permet typiquement d'étendre la connectivité à une passerelle Internet en assurant la continuité de service. De même, les réseaux ad-hoc maillés [105] permettent de construire le cœur d'un réseau d'interconnexion en reliant classiquement les points d'accès sans fil des habitations dans un quartier résidentiel. La multiplication des applications fait naître de nouveaux besoins en termes de gestion qui sont à l'origine de nouveaux travaux de recherche.

1.2 Problématique

La supervision regroupe un ensemble d'activités qui permet de surveiller et contrôler les réseaux et leurs services. Elle est aujourd'hui confrontée à des environnements de plus en plus dynamiques dont les réseaux ad-hoc en sont un des exemples les plus caractéristiques. Les architectures de gestion traditionnelles, initialement conçues pour les infrastructures fixes, sont inadaptées à la nature dynamique et aux contraintes fortes des réseaux ad-hoc. Ainsi, elles prennent difficilement en charge les changements fréquents de topologie du réseau et sont souvent trop consommatrices en ressources dans un contexte où bande passante et énergie sont fortement limitées. De nouveaux verrous scientifiques et techniques doivent donc être levés pour intégrer les réseaux ad-hoc dans une démarche de gestion.

La problématique de mes travaux de recherche porte sur une nouvelle approche de gestion pour les réseaux et services ad-hoc capable de prendre en compte leur nature dynamique et leurs ressources limitées. Cette approche doit être facilement intégrable aux infrastructures de gestion actuelles, suffisamment flexible pour s'adapter dynamiquement aux changements au sein du réseau ad-hoc, et suffisamment économe pour limiter la charge induite par l'activité de gestion sur le fonctionnement même du réseau. Ce travail de recherche s'organise autour de trois axes majeurs qui correspondent respectivement à (1) la construction d'un modèle d'information générique pour les réseaux ad-hoc, (2) une réorganisation plus souple du plan de gestion à partir d'une méthode probabiliste et enfin (3) l'adaptation des opérations de gestion, dans le contexte de la gestion de performances en utilisant des techniques de filtrage, et dans le contexte de la

gestion de fautes en s'appuyant sur la théorie de l'information.

Les travaux de recherche présentés dans ce manuscript ont été réalisés dans le cadre de ma thèse au sein de l'équipe MADYNES dirigée par Olivier Festor au LORIA/INRIA Lorraine à Nancy. Celle-ci s'est déroulée sous la direction d'André Schaff et de Radu State et s'inscrit dans la thématique plus générale de notre équipe qui porte sur la gestion de réseaux et services dynamiques. Elle a été partiellement effectuée dans le cadre du projet RNRT SAFARI [64].

1.3 Organisation du manuscript

Le manuscript est composé de neuf chapitres organisés en trois parties principales ainsi que d'une annexe. Les trois parties correspondent respectivement à (1) l'état de l'art de notre domaine de recherche, (2) la présentation de nos différentes contributions et (3) la description de leur mise en œuvre à travers un prototypage.

1.3.1 Partie I : Etat de l'art

La première partie constitue une introduction à la gestion des réseaux et services ad-hoc dans laquelle sont dépeints les principaux modèles et architectures ainsi que leurs champs d'applications. Elle relève les limites théoriques et pratiques des approches actuelles et nous sert de support pour positionner nos travaux de recherche.

Modèles et architectures de gestion

Le chapitre 2 présente tout d'abord les modèles et architectures de gestion à destination des réseaux ad-hoc. Nous rappelons brièvement les concepts de base de la gestion de réseau et ses objectifs, avec une description des quatre modèles génériques sur lesquels s'appuie toute infrastructure : modèle de l'information, modèle organisationnel, modèle de communication et modèle fonctionnel. Puis, nous mettons en évidence les propriétés des réseaux ad-hoc afin d'identifier les défis posés par leur intégration dans un cadre de gestion. La topologie dynamique aboutit par exemple à réorganiser le plan de gestion. L'absence d'infrastructure fixe pousse au déploiement de mécanismes d'auto-gestion, et l'hétérogénéité des équipements favorise la construction d'un socle commun de protocoles et de services. Nous détaillons les différentes infrastructures de gestion sous-jacentes en précisant leurs potentiels et leurs limites.

Domaines d'applications

Le chapitre 3 décrit ensuite les différents domaines d'applications de la supervision dans ce contexte. Nous y analysons les approches de monitorage qui permettent d'observer le réseau ad-hoc, de mesurer son état de fonctionnement et de distinguer la part des ressources consommées par un nœud ad-hoc. Nous détaillons ensuite les approches de configuration qui permettent le paramétrage du réseau ad-hoc et de ses entités, en nous intéressant en particulier à la configuration des adresses IP à travers différentes politiques : configuration sans conflit, configuration avec détection de conflits et configuration *best effort*. Nous décrivons enfin les mécanismes de contrôle qui permettent d'assurer à différents niveaux un partage équitable des ressources offertes par le réseau : contrôle de l'accès au médium radio, contrôle du plan de routage et contrôle de services.

1.3.2 Partie II : Contributions

La seconde partie présente nos travaux de recherche portant sur une nouvelle approche intégrée, flexible et économe pour la gestion des réseaux et services ad-hoc. Celle-ci se traduit par la construction d'un modèle d'information, la réorganisation du plan de supervision et l'adaptation des opérations de gestion. Cette partie inclut de nombreux résultats expérimentaux obtenus par la simulation. Ceux-ci permettent l'évaluation de nos travaux et sont complétés par des résultats analytiques.

Modélisation étendue de l'information de gestion

Le chapitre 4 décrit la modélisation de l'information de gestion à travers l'identification des éléments caractéristiques d'un réseau ad-hoc et leurs spécifications à l'aide d'un formalisme commun. Les approches de gestion dédiées aux réseaux ad-hoc négligent le modèle d'information en omettant de le définir ou en le définissant de manière très partielle. Ce modèle est pourtant essentiel car il fournit un cadre formel pour la description des ressources gérées et la structuration de l'information de gestion. Nous définissons notre modèle de manière générique sous la forme d'une extension du modèle commun de l'information (CIM) [39]. Il prend notamment en considération l'organisation du réseau ad-hoc, les échanges en son sein à différentes échelles et la participation des nœuds à son bon fonctionnement. Nous introduisons également un sous-modèle permettant la prise en charge du protocole de routage ad-hoc OLSR [57].

Organisation probabiliste du plan de gestion

Le chapitre 5 définit une nouvelle organisation du plan de gestion à partir d'une méthode probabiliste. Une démarche de gestion au sens pur du terme, où l'ensemble des nœuds serait géré à tout moment, est trop stricte pour les réseaux ad-hoc. Au lieu de considérer la gestion du réseau dans son intégralité, nous relâchons les contraintes sur le plan de gestion en ne considérant que certains nœuds, ceux qui disposent à la fois d'une forte présence dans le réseau et d'une forte connectivité avec leur voisinage. A partir de cette approche sélective, nous dérivons des garanties sur le pourcentage de nœuds qui seront actifs dans le plan de gestion. Nous détaillons la méthode algorithmique considérée en définissant une mesure de connectivité spatio-temporelle, l'extraction de composantes spatio-temporelles et le déploiement de mécanismes électifs utilisant la centralité de degré et la centralité par vecteur propre. Nous montrons comment cette méthode peut être intégrée au sein de l'architecture de gestion ANMP [53].

Gestion de performances par filtrage et analyse de graphes

Le chapitre 6 présente ensuite une adaptation de la gestion de performances. Nous définissons de nouvelles méthodes d'analyse permettant de construire une vue fonctionnelle synthétique du réseau et de déterminer l'impact des nœuds sur son fonctionnement. Une technique de filtrage compare l'état des nœuds dans un voisinage local tandis que l'analyse de graphes met en évidence les dépendances entre ceux-ci. La stratégie permet de faire apparaître les chemins qui sont les plus utilisés lors de communications multi-sauts et qui représentent en fait les *backbones* du réseau ad-hoc. Elle permet aussi de quantifier l'impact des nœuds sur le fonctionnement global du réseau en mettant en évidence les disparités entre nœuds. Cet impact peut être positif (par exemple, les nœuds qui ont une activité de routage importante et qui font partie d'un *backbone*) ou négatif (par exemple, les nœuds qui refusent d'intervenir comme routeurs ou qui consomment

abusivement les ressources). Les données issues de cette observation peuvent être exploitées à des fins de reconfiguration, de positionnement de sondes et de provisionnement de nœuds.

Gestion de fautes par inférence

Dans une démarche similaire, le chapitre 7 porte sur une adaptation de la gestion de fautes. Si la gestion de fautes est un problème bien connu dans les réseaux fixes classiques, la nature fortement dynamique des réseaux ad-hoc amène à repenser cette activité. Nous définissons une méthode de gestion qui consiste à analyser l'intermittence des nœuds ad-hoc et à détecter des fautes/pannes par inférence. L'intermittence d'un nœud peut être provoquée par des causes bénignes telles que la mobilité ou la dégradation temporaire de la connectivité. Elle devient cependant pathologique lorsqu'elle est causée par des erreurs de configuration, des pannes de routage, des problèmes de batterie. Un problème majeur consiste à différencier une intermittence pathologique d'une intermittence régulière pour pouvoir ainsi détecter les nœuds non opérationnels. Nous proposons pour ce faire une mesure fondée sur la théorie de l'information permettant de caractériser l'intermittence d'un nœud. Nous introduisons différentes méthodes collaboratives de détection, incluant un mécanisme d'auto-configuration, pour identifier les nœuds pathologiques de manière distribuée.

1.3.3 Partie III : Mise en œuvre

Le chapitre 8 présente enfin la mise en œuvre des différentes contributions à travers le développement d'une plate-forme de monitorage. Cette mise en œuvre complète les différents travaux de simulations et d'analyses par une expérimentation pratique. Le développement de la plate-forme comprend le prototypage d'un gestionnaire et d'un agent-sonde ainsi que le développement d'un module d'analyses statistiques. Ce module nous a permis d'expérimenter nos méthodes de gestion, comme par exemple l'application de techniques de filtrage, l'analyse de la centralité des nœuds du réseau et l'étude de la distribution de paquets dans le plan de routage.

Enfin, la conclusion donne un récapitulatif des contributions réalisées durant ces trois années de thèse et expose un ensemble de perspectives de recherche.

Première partie

Confrontation de la supervision aux environnements ad-hoc

Chapitre 2

Modèles et architectures de gestion

Sommaire

2.1 Introduction

Parallèlement à la croissance des infrastructures réseaux et à la multiplication des services offerts, la gestion[1] de réseaux et services est confrontée à des environnements de plus en plus dynamiques tels que les réseaux ad-hoc. Ceci nécessite de définir de nouveaux modèles et architectures de gestion capables de s'adapter au changement et de prendre en compte des contraintes supplémentaires. Dans ce chapitre, nous aborderons tout d'abord les concepts de base de la gestion de réseaux en décrivant le processus de gestion et en définissant les quatre principaux modèles sur lesquels s'appuie toute approche de gestion. Nous présenterons ensuite les réseaux ad-hoc et montrerons dans quelles mesures leurs caractéristiques conduisent à de nouveaux défis en termes de gestion. Enfin, nous détaillerons, à proprement parler, les principales approches de gestion mises en œuvre pour les réseaux ad-hoc.

[1] Nous emploierons indifféremment dans ce manuscript les termes de gestion, supervision et administration.

2.2 Concepts de la gestion de réseaux et services

Gérer les réseaux et services [76] est une activité essentielle pour toute infrastructure qui fournit des services au sein d'un réseau informatique. Elle a pour objectif la surveillance et le contrôle des éléments matériels et logiciels du réseau afin de répondre aux exigences des usagers. Celles-ci sont souvent spécifiées sous la forme d'objectifs fixés dans le cadre de politiques et de contrats de services. Dans un cadre général, il est possible de définir la gestion de réseaux et de services de la manière suivante.

Définition 1 *La gestion de réseaux et services regroupe l'ensemble des activités qui consiste à mettre en œuvre, maintenir et mettre à jour les infrastructures de réseaux et leurs services afin que ceux-ci respectent les contraintes de qualité et de coût spécifiées dans les objectifs et les contrats de services.*

Les contraintes de coût intègrent également les coûts liés à l'activité de gestion elle-même. La gestion de réseaux implique la consommation de ressources et ne doit pas empêcher le bon fonctionnement du réseau.

2.2.1 Processus de gestion

L'activité de gestion peut être representée sous la forme d'un processus [106] tel qu'illustré en figure 2.1. La gestion consiste à établir une boucle de contrôle autour d'un système initialement libre afin de maintenir le niveau de service demandé fixé par les objectifs $\{s_1, s_2, ..., s_q\}$.

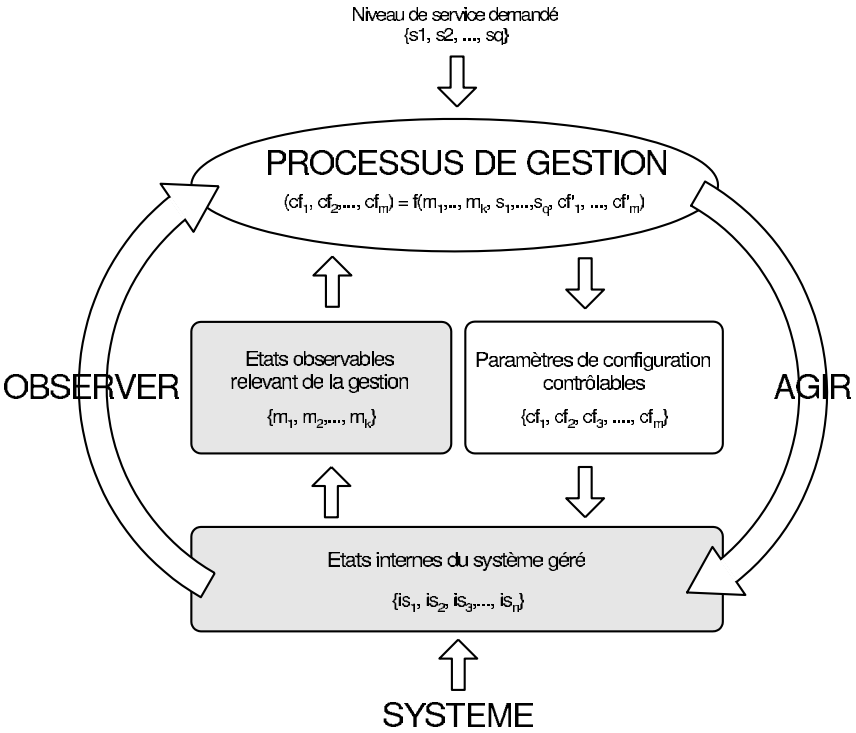


FIG. 2.1 – Modélisation du processus de gestion

L'intégration d'une infrastructure de supervision va consister en deux activités de base : observer et agir sur le système. Les états internes $\{is_1, is_2, ..., is_n\}$ du système ne sont pas tous

directement accessibles. La première activité a pour but d'observer le système à travers les états observables $\{m_1, m_2, ..., m_k\}$ afin d'en établir une vue abstraite. Elle peut être accomplie soit par une surveillance régulière du système par le biais de rapports et de sondages périodiques, soit par la mise en place d'alarmes qui sont enclenchées lors de l'apparition d'anomalies. A partir de cette observation, la seconde activité consiste à agir sur le système en influant sur les paramètres de configuration $\{cf_1, cf_2, ..., cf_m\}$. Cette activité ne se limite pas aux réglages du système mais intègre également à travers cette représentation de nombreuses actions telles que la mise en œuvre, la maintenance, la mise à jour du système. Le processus de gestion représente une fonction qui prend en entrée à la fois le niveau de service demandé, les états observables et éventuellement les anciens paramètres de configuration $\{cf'_1, cf'_2, ..., cf'_m\}$. Il retourne en sortie les paramètres de configuration adéquats $\{cf_1, cf_2, ..., cf_m\}$ à appliquer au système géré. A travers cette boucle de contrôle, nous pouvons constater que la gestion se rapproche du concept d'asservissement d'un système en automatique.

2.2.2 Modèles de la gestion de réseaux

Une infrastructure de gestion de réseaux s'appuie sur une modélisation multi-dimensionnelle [104]. On considère classiquement quatre modèles correspondant à une abstraction et à des objectifs différents. Nous allons définir chacun de ces modèles et nous les illustrerons en présentant brièvement les standards associés.

Modèle de l'information

Le modèle de l'information définit un cadre formel commun pour la description des ressources gérées et la structuration de l'information de gestion. Il doit offrir un niveau d'abstraction suffisant pour offrir une vue homogène et extensible de l'ensemble des ressources, et ce quelles que soient la nature, la localisation et les méthodes d'accès de celles-ci. Une ressource est typiquement modélisée par un objet dont les attributs en indiquent l'état et sont accessibles par des opérations de gestion. La représentation homogène des ressources facilite la définition d'un service générique permettant d'accéder à cette information.

Parmi les premières approches standards, la gestion SNMP [48] spécifiée par l'IETF propose la représentation des objets administrés sous la forme d'une structure relativement simple composée d'un ensemble de variables qui sont stockées dans une base d'information de gestion appelée MIB (*Management Information Base*) [159]. Cette structure est décrite à l'aide d'un langage d'expression formelle appelé SMIv2 [137] .

Le modèle d'information CIM [39] introduit par le DMTF [2] à travers l'initiative WBEM [135] définit, quant à lui, une solution réellement fondée sur le paradigme objet mais est parfois plus complexe à mettre en œuvre. Il représente les ressources à gérer sous une forme graphique similaire à celle d'un langage de modélisation objet tel qu'UML [65], et s'appuie sur le langage MOF (*Managed Object Format*) comme langage d'expression formel.

Modèle organisationnel

Le modèle organisationnel décrit le rôle et les relations de chacune des entités intervenant dans la tâche de gestion. L'organisation du plan de gestion s'appuie sur le concept gestionnaire/agent. Le gestionnaire est l'entité responsable de l'activité de gestion et assure l'interface avec l'utilisateur : il émet des requêtes d'opérations auprès d'un ou plusieurs agents. Symétriquement, l'agent assure l'interface avec les ressources gérées : il exécute les requêtes d'opérations, envoie des réponses au gestionnaire et peut émettre des notifications d'événements.

Plusieurs types d'organisation de gestion [178, 115] peuvent être définis à partir de ce concept. On distingue généralement quatre types d'organisation différents qui sont présentés en figure 2.2 :

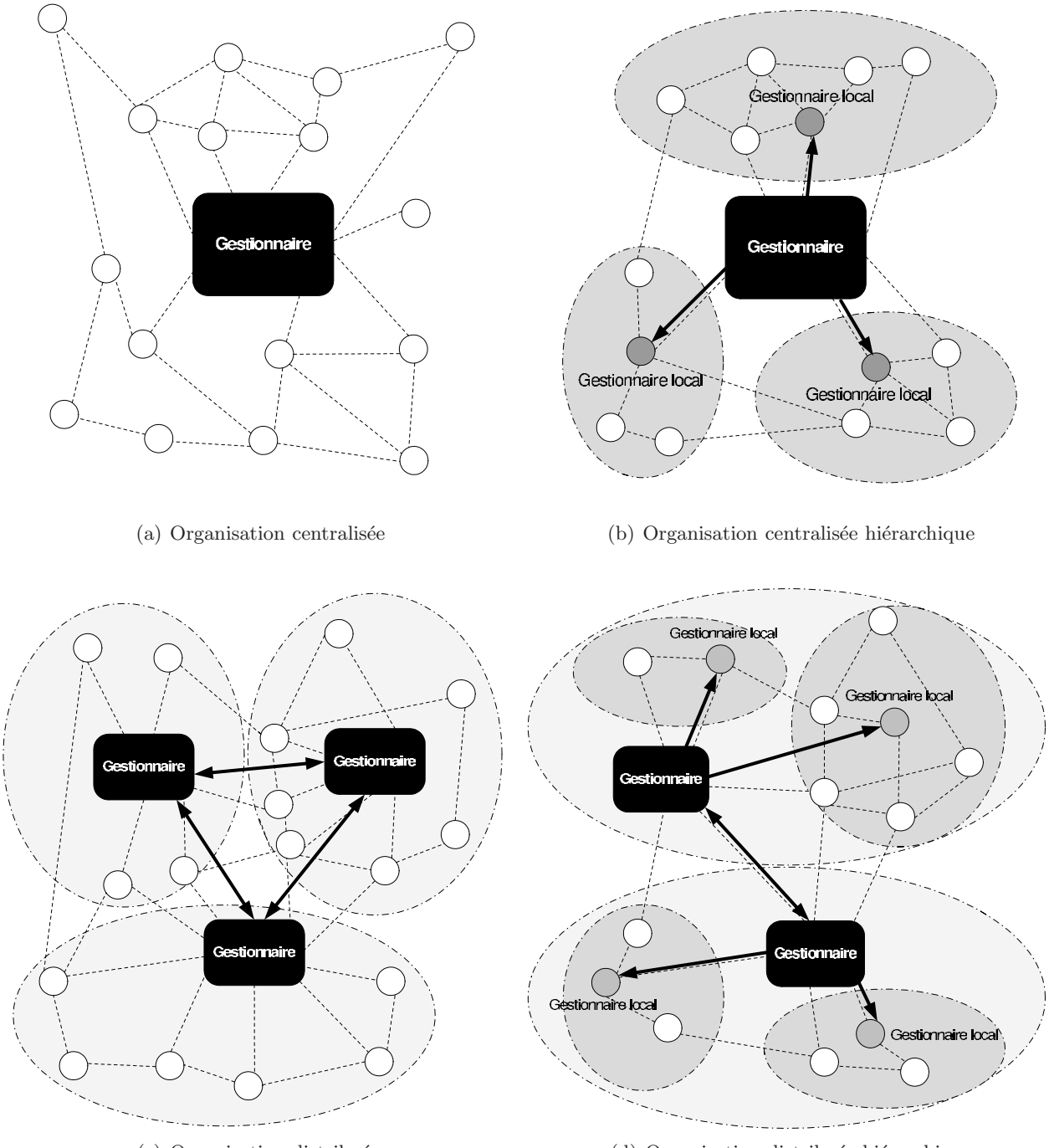
- **organisation centralisée** (figure 2.2(a)) : cette organisation repose sur un unique gestionnaire chargé de collecter les données auprès des agents et de contrôler l'ensemble du réseau. On la retrouve dans les premières infrastructures de gestion de réseaux. Elle offre en effet une implantation facile en ne considérant qu'un seul point de contrôle. Les limites de cette approche sont liées à une dépendance forte due à la centralisation exclusive des opérations de gestion. D'une part, cette organisation n'est pas robuste puisque l'ensemble de l'infrastructure s'écroule si le gestionnaire n'est plus opérationnel. D'autre part, elle génère un trafic de gestion important de et vers le gestionnaire car toutes les opérations de gestion sont opérées par cette unique entité.
- **organisation centralisée hiérarchique** (figure 2.2(b)) : cette organisation conserve une autorité centrale mais introduit une hiérarchie de gestionnaires locaux afin d'assurer un meilleur passage à l'échelle. Le gestionnaire utilise des gestionnaires locaux comme intermédiaires afin de répartir les opérations de gestion. Chaque gestionnaire dispose d'un certain niveau de responsabilité dans la tâche de gestion. Un gestionnaire local est responsable d'un sous-ensemble du réseau dont il collecte les données de gestion et peut les transmettre si nécessaire à un gestionnaire d'un plus haut degré de responsabilité. Le gestionnaire central reste l'unique point de contrôle et dispose donc du plus haut degré de responsabilité.
- **organisation distribuée** (figure 2.2(c)) : la gestion est réalisée de façon distribuée par un ensemble de gestionnaires qui communiquent entre eux. Les gestionnaires disposent d'un même degré de responsabilité et sont chacun pleinement responsable de la gestion d'un sous-ensemble de nœuds dans le réseau. L'utilisation de gestionnaires multiples permet de limiter localement la charge de gestion et d'améliorer la robustesse du système. En revanche, des mécanismes de coopération doivent être mis en œuvre pour assurer la cohérence dans les opérations de gestion exécutées par les gestionnaires.
- **organisation distribuée hiérarchique** (figure 2.2(d)) : il s'agit d'une approche distribuée où chaque gestionnaire peut déléguer une partie des tâches de gestion à des gestionnaires locaux. Un gestionnaire n'a d'autorité que sur les gestionnaires locaux de son domaine, à moins que des mécanismes de délégation soient mis en œuvre.

Modèle de communication

Le modèle de communication spécifie le protocole d'échanges des informations de gestion entre les différentes entités. Il s'agit notamment d'assurer les échanges au niveau applicatif entre le gestionnaire et l'agent afin de transmettre les opérations et permettre l'accès et la manipulation des données de gestion issues d'une base d'information.

Le protocole SNMP est le standard déployé dans la majorité des infrastructures de gestion des réseaux actuels. Il repose sur des messages de type requête et réponse : le gestionnaire émet une requête de service vers un agent et l'agent transmet en retour une réponse au gestionnaire. L'agent peut également émettre une notification sans attendre de retour de la part du gestionnaire. Le service fourni par ce protocole est constitué de cinq primitives qui permettent de lire des informations sur des objets gérés dans la base d'informations de gestion, de mettre à jour cette base d'informations et d'émettre des notifications suite à des événements.

De nouveaux protocoles qui s'appuient sur le langage XML [198] connaissent un essor important ces dernières années. En particulier, le protocole NetConf [73] défini à l'IETF assure à la fois l'encodage des données de gestion et la définition des messages du protocole à l'aide du



(a) Organisation centralisée

(b) Organisation centralisée hiérarchique

(c) Organisation distribuée

(d) Organisation distribuée hiérarchique

Fig. 2.2 – Types d'organisation pour la gestion de réseaux et services

langage XML [198]. Les messages de gestion, spécifiés sous la forme de schémas XML [188], sont transmis au dessus du protocole RPC [31] d'appel de procédures à distance. L'usage de XML facilite l'interopérabilité des systèmes de gestion mais peut alourdir de façon significative le coût de gestion en terme de trafic réseau.

Modèle fonctionnel

Ce modèle permet de répartir les opérations de gestion par aire fonctionnelle. La classification de l'ISO[2] [78] définit cinq aires principales usuellement connues sous l'acronyme FCAPS [76] et décrites ci-dessous :

- **gestion des fautes** : cette aire fonctionnelle a pour but la détection, l'isolation et la correction des anomalies qui affectent le fonctionnement des réseaux et de leurs services. Ces fautes peuvent être causées par la panne d'un équipement physique aussi bien que par un dysfonctionnement d'origine logicielle. La gestion de fautes vise à minimiser l'impact des fautes sur les services tout en limitant les interférences induites par les opérations de détection et de correction elles-mêmes.
- **gestion de la configuration** : elle comprend le recensement des ressources du réseau ainsi que leur configuration physique et logicielle. Elle intervient notamment lors de l'intégration de nouveaux équipements ou lors du déploiement de nouveaux services à travers les différentes opérations de configuration.
- **gestion de la comptabilité (*accounting*)** : elle consiste à évaluer les usages des infrastructures et des services du réseau ainsi qu'à comptabiliser les coûts relatifs. Elle comprend en outre l'établissement de rapports de consommations ainsi que les activités liées à la facturation dans le cas de services payants.
- **gestion de la performance** : son objectif est d'évaluer la qualité du service délivrée par le réseau et de la maintenir grâce à des opérations de contrôle. Elle comprend les opérations de monitorage qui permettent de déterminer l'état de fonctionnement du réseau à travers différents critères de qualité tels que la disponiblité de service, mais aussi les opérations de prévention et de correction qui permettent de garantir le niveau de performances souhaité.
- **gestion de la sécurité** : elle vise la protection du réseau en empêchant l'ensemble des activités frauduleuses qui peuvent avoir un impact sur l'intégrité et le bon usage des services. Elle comprend les mécanismes d'authentication, de contrôle d'accès et de confidentialité ainsi que l'administration des infrastructures de sécurité.

A noter que si ces quatre modèles sont présents dans toute infrastructure de gestion, ils n'ont généralement pas la même importance relative au sein d'une même approche.

2.3 Intégrer les réseaux ad-hoc dans une démarche de gestion

Les avancées importantes dans les technologies sans fil ont favorisé le développement de réseaux de plus en plus pervasifs qui visent à fournir aux utilisateurs de la connectivité pratiquement en tout lieu et à tout instant. Le nombre de réseaux et d'équipements mobiles a en conséquence considérablement augmenté ces dernières années. Cependant, les approches de gestion, initialement dédiées à des infrastructures fixes, sont souvent inadaptées à la mobilité de ces réseaux, notamment dans le cas de réseaux fortement décentralisés tels que les réseaux mobiles ad-hoc.

[2]International Standards Organization

2.3.1 Principe des réseaux ad-hoc

Les réseaux ad-hoc [157, 147] sont apparus dans les années soixante-dix. Ils correspondent à une catégorie de réseaux mobiles qui peuvent fonctionner sans infrastructure fixe en établissant des communications directes ou multi-sauts [128] entre les équipements. Nous pouvons caractériser de manière générale un réseau mobile ad-hoc par la définition suivante.

Définition 2 *Un réseau mobile ad-hoc (MANET, Mobile Ad-Hoc Network) est un réseau auto-organisé formé spontanément à partir d'un ensemble d'entités mobiles communicantes (ordinateurs portables, téléphones mobiles, assistants électroniques) sans nécessiter d'infrastructure fixe préexistante.*

Contrairement aux réseaux cellulaires qui requièrent un important effort de planification pour leurs déploiements, les réseaux ad-hoc peuvent être déployés facilement et rapidement en permettant des échanges directs entre stations mobiles. Ainsi, le fonctionnement du réseau repose sur les stations mobiles elles-mêmes : celles-ci interviennent à la fois comme terminaux pour communiquer avec les autres usagers et comme routeurs afin de relayer le trafic pour le compte d'autres utilisateurs. En particulier, lorsque deux stations ne peuvent communiquer directement parce qu'elles ne se trouvent pas dans le même voisinage, une communication multi-sauts est initialisée en utilisant les nœuds intermédiaires : les paquets sont transmis de station en station avant d'atteindre la destination souhaitée.

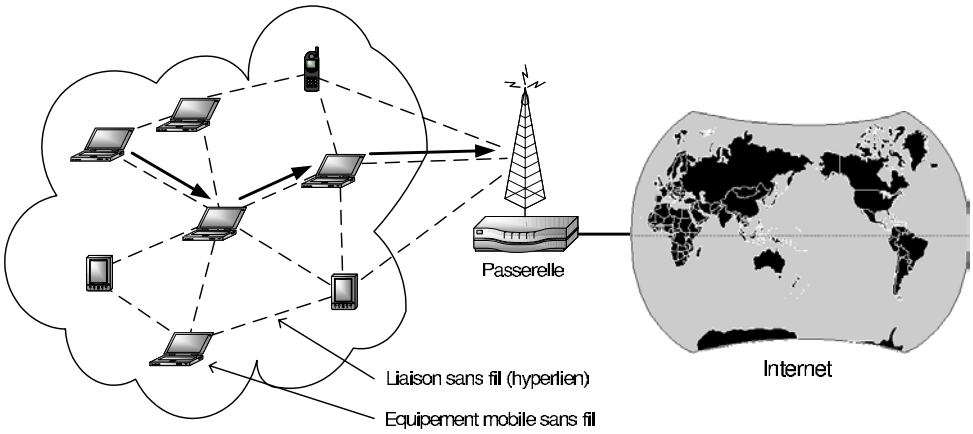


Fig. 2.3 – Scénario d'utilisation d'un réseau ad-hoc

Les réseaux ad-hoc permettent d'offrir de la connectivité avec un déploiement rapide et à faible coût dans différents domaines d'applications : services de secours, communications inter-véhiculaires, boucles locales radio, réseaux en environnements hostiles. Bien qu'ils ne nécessitent pas d'infrastructure fixe, les réseaux ad-hoc sont souvent liés dans la pratique à une passerelle permettant l'interconnexion à l'Internet filaire, comme décrit à la figure 2.3. Si l'on considère le scénario d'un opérateur de télécommunications fournissant un accès à l'Internet à travers un ou plusieurs points d'accès sans fil, le déploiement d'un réseau ad-hoc peut être justifié typiquement dans les situations suivantes :

– l'opérateur doit couvrir une zone de très faible densité (le coût d'installation de points d'accès s'avère élevé par rapport à la population qui est prête à souscrire au service) ou de très forte densité (le coût du câblage de l'ensemble de la population est important),

– l'opérateur doit fournir un service dans un environnement où le déploiement des points d'accès est techniquement difficile à réaliser (installation de points d'accès en régions montagneuses),
– le réseau fixe est endommagé (panne d'un point d'accès, catastrophes naturelles), les clients souhaitent continuer à disposer du service.

Dans les situations énoncées, les terminaux des utilisateurs passent en mode ad-hoc afin de communiquer entre eux par communications directes et multi-sauts : les nœuds du réseau interviennent en tant que routeurs pour (1) assurer la liaison entre des nœuds qui ne sont pas voisins directs et (2) permettre aux nœuds, qui n'ont pas de points d'accès (fonctionnels) dans leur voisinage, d'accéder à l'Internet en passant par des nœuds intermédiaires. L'émergence de nouvelles applications est favorisée par les avancées dans les technologies physiques (miniaturisation des composants, réduction de la consommation énergétique).

2.3.2 Caractéristiques des réseaux ad-hoc

Comparativement aux réseaux fixes traditionnels, les réseaux ad-hoc permettent de réduire considérablement les coûts aussi bien de déploiement que de maintenance des infrastructures grâce à une auto-organisation du réseau. Ils présentent en contre partie des contraintes additionnelles :

– **topologie dynamique** : les réseaux ad-hoc sont formés spontanément à partir de nœuds mobiles sans nécessiter l'appui d'une infrastructure fixe. Un nœud ad-hoc est susceptible de quitter ou de rejoindre le réseau à tout instant. La topologie du réseau peut donc être fortement changeante au grès des déplacements des nœuds et de leur état de fonctionnement [23, 179]. Ceci a un impact sur le protocole de routage qui doit assurer une maintenance régulière des chemins.
– **hétérogénéité des nœuds** : les nœuds ad-hoc peuvent correspondre à une multitude d'équipements : de l'ordinateur portable au capteur intelligent en passant par le téléphone mobile. Ces équipements ne disposent pas des mêmes propriétés physiques et logicielles, mais elles doivent pourtant interopérer pour établir un réseau commun [187].
– **bande passante limitée** : le support physique sans fil offre une bande passante limitée qui doit être partagée entre les nœuds d'un même voisinage [89, 87]. La bande passante disponible dépend à la fois du nombre de nœuds présents dans le voisinage et du trafic de données à transporter, indépendamment des perturbations physiques qui peuvent intervenir.
– **contraintes d'énergie** : les équipements mobiles sont fortement contraints par la durée de vie limitée de leurs batteries. Malgré les améliorations des batteries et des technologies de plus en plus économes en énergie, les nœuds ad-hoc sont particulièrement sollicités notamment lorsqu'ils relayent le trafic réseau pour le compte d'autres nœuds [186, 163].
– **sécurité limitée** : la nature du support physique ainsi que l'absence de coordination centrale rendent les réseaux ad-hoc plus vulnérables que les infrastructures fixes. Les transmissions sans fil peuvent être aisément capturées par un nœud ad-hoc dans le voisinage local. Une attaque par déni de service peut être facilement réalisée par un nœud malicieux en s'appropriant la bande passante ou en surchargeant un nœud voisin avec une quantité importante de trafic à router [41, 184].

Le développement des réseaux ad-hoc repose essentiellement sur la standardisation de protocoles de routage ad-hoc adaptés à ces caractéristiques.

2.3.3 Protocoles de routage ad-hoc

La standardisation des protocoles ad-hoc s'opère essentiellement au niveau de la couche réseau IP afin d'offrir une meilleure interopérabilité entre les réseaux et de permettre des technologies hétérogènes dans les couches basses. Une multitude de protocoles de routage a été développée spécifiquement pour les réseaux ad-hoc ces dernières années. L'absence de coordination centrale et de stations de base rend la tâche de routage plus complexe que dans les réseaux fixes. De plus, le plan de routage doit être capable de s'adapter à la mobilité de ces réseaux tout en consommant le moins de ressources possibles car les capacités des terminaux sont généralement fortement contraintes. Le groupe de travail MANET [132] de l'IETF a en charge la standardisation des protocoles de routage à destination des réseaux ad-hoc. Ces protocoles peuvent être classifiés en fonction du mécanisme de mise à jour des informations de routage : routage réactif et routage proactif.

- **routage réactif** : le protocole établit une route uniquement lorsqu'elle est demandée par un nœud mobile, en initialisant un mécanisme de découverte. Dans une approche réactive, les nœuds mobiles ne maintiennent pratiquement pas d'informations sur la topologie du réseau. Le nœud source émet une requête de route qui est diffusée dans le réseau ad-hoc jusqu'au nœud destination. Les nœud intermédiaires sont découverts et mémorisés durant cette phase de diffusion. Lorsque le nœud destination reçoit la requête, il utilise le chemin inverse pour contacter le nœud source et lui transmettre les informations de routage. Un exemple typique de protocole réactif est le protocole AODV (*Ad-hoc On-Demand Distance Vector*) spécifié par la RFC 3561 [156].
- **routage proactif** : l'approche proactive consiste, à l'inverse, à ce que chaque nœud du réseau maintienne une table de routage. Les mises à jour de la table sont obtenues par échange périodique d'informations de topologie entre les nœuds. Un exemple typique de protocole proactif est le protocole OLSR (*Optimized Link State Routing Protocol*) spécifié dans la RFC 3626 [57]. Ce protocole est une optimisation des protocoles à état de liens adaptés à la nature des réseaux ad-hoc. Chaque nœud réalise deux opérations principales : il détermine la liste des voisins directs en évaluant le voisinage par émission périodique de messages HELLO et il échange les information de topologie avec les autres nœuds en diffusant des messages TC de contrôle de topologie. L'heuristique du protocole OLSR permet de réduire la charge de trafic durant la phase de diffusion en ne sélectionnant qu'un sous-ensemble de nœuds appelés relais multi-points pour la retransmission des messages de contrôle.

Ces deux approches de routage ne présentent pas les mêmes propriétés [38] et le choix de l'une ou l'autre doit être réalisé en fonction du type d'applications envisagé. Un protocole réactif permet un meilleur passage à l'échelle car il réduit de manière significative la charge de trafic dans le réseau puisque les informations de topologie ne sont pas diffusées périodiquement. Cependant, une approche proactive offre un meilleur temps de latence pour établir une route puisqu'elle ne nécessite aucun mécanisme de découverte au préalable. Des protocoles de **routage hybride** tels que le protocole ZRP (*Zone Routing Protocol*) [99] combinent proactivité et réactivité. Ils permettent d'offrir un compromis entre charge de trafic et temps de latence. La topologie des nœuds qui se trouvent dans une zone proche est maintenue dans une table de routage à travers une approche proactive. *A contrario*, les nœuds situés en dehors de cette zone de voisinage (à plus d'un certain nombre de sauts) sont atteignables par découverte de route grâce à une approche réactive.

Le plan de routage converge progressivement vers des protocoles standardisés, l'objectif étant toujours de définir des solutions capables de s'adapter aux contraintes fortes imposées par les

réseaux ad-hoc. Pour sa part, le plan de gestion reste encore insuffisamment exploré.

2.3.4 Défis à relever en termes de gestion

Les modèles et architectures de gestion usuellement utilisés pour les infrastructures fixes ne peuvent être directement appliqués dans le cadre des réseaux ad-hoc. De nouvelles approches doivent être mises en œuvre pour répondre à la dynamique et aux contraintes de ces réseaux. Ceux-ci nous confrontent à de multiples défis à relever en termes de gestion, parmis lesquels :

- **spécifier l'information de gestion** : les réseaux ad-hoc représentent un nouvel environnement dont il est nécessaire d'identifier les ressources à gérer ainsi que les opérations de gestion qui sont possibles à entreprendre. L'enjeu consiste à définir les informations essentielles qui relèvent du contexte ad-hoc, puis à les structurer dans un formalisme commun. Ce formalisme est essentiel pour garantir une démarche de gestion homogène quelles que soient la nature du réseau et les technologies utilisées.
- **délimiter le domaine administratif** : ces réseaux sont formés dynamiquement à partir d'équipements faiblement couplés, si bien que la notion de domaine administratif est plus délicate à définir. Même si les approches coopératives et à base de rôle, où les rôles de gestionnaires et d'agents sont assignés dynamiquement, sont possibles d'un point de vue technique, la gestion de réseaux a besoin de définir des politiques métier à un niveau d'abstraction plus élevé pour les appliquer à des domaines administratifs.
- **organiser le plan de supervision** : dans la mesure où les réseaux ad-hoc sont des réseaux auto-organisés évoluant dynamiquement dans le temps, le plan de supervision doit disposer de propriétés similaires d'auto-organisation lui permettant de s'adapter aux changements du réseau. L'architecture de gestion doit pouvoir être déployée de la manière la plus autonome possible en limitant les interventions manuelles de configuration. Elle doit également prendre en considération les caractéristiques hétérogènes des nœuds lors de l'affectation des tâches de gestion.
- **minimiser les coûts de gestion** : les ressources en bande passante et en énergie sont fortement contraintes dans cet environnement. Les opérations de gestion sont consommatrices en ressources parce qu'elles génèrent un trafic supplémentaire au sein du réseau et exploitent localement les ressources énergétiques et calculatoires des nœuds. Définir des approches de gestion légères voire passives, où les informations déjà disponibles sont exploitées autant que possible, permettent de réduire cette consommation et d'optimiser en conséquence la durée de vie des nœuds et du réseau.
- **fiabiliser les données de gestion** : un nœud ad-hoc n'est pas fiable à cause de ses ressources internes limitées (pannes de batterie) et de son environnement extérieur incertain (perte de connectivité due à des perturbations physiques), si bien que les données de gestion ont une probabilité importante de se dégrader ou d'être perdues. Un nœud non coopératif ou malicieux peut également refuser de fournir des données ou les falsifier. Il est nécessaire que les modèles de gestion soient capables de fiabiliser les données de gestion en dévaluant les informations erronées.
- **offrir une meilleure robustesse** : la mobilité amène un réseau ad-hoc à se partitionner ou à fusionner avec d'autres réseaux. Le plan de supervision doit rester cohérent et être facilement maintenu même lors de fusions/partitions du réseau. Les protocoles de routage ad-hoc prennent en charge des réseaux de plus en plus larges, de sorte que l'infrastructure de gestion doit être suffisamment robuste pour supporter les facteurs d'échelle.

Les paradigmes de gestion qui permettront de répondre à ces défis devront par ailleurs assurer la convergence vers une démarche commune de gestion pour les réseaux ad-hoc et filaires.

2.4 Approches de gestion pour les réseaux ad-hoc

Nous allons présenter les principales approches de gestion qui ont été proposées ou redéfinies pour les réseaux ad-hoc et en détaillerons les architectures. Ces approches reposent sur des paradigmes qui ne sont pas nécessairement nouveaux mais qui trouvent leur pleine expression dans le contexte de réseaux fortement dynamiques.

2.4.1 Gestion par délégation

La gestion par délégation permet de déléguer les opérations de gestion effectuées par le gestionnaire à des agents [197, 96]. Les opérations de gestion impliquent de nombreux échanges entre le gestionnaire et les agents. Le gestionnaire est souvent amené à exécuter plusieurs opérations de gestion simultanément en mobilisant des ressources importantes. Le mécanisme de délégation permet de décharger le gestionnaire en déléguant une partie des opérations de gestion à ses agents, typiquement en considérant des gestionnaires locaux. L'idée est de décentraliser la fonction de gestion en augmentant l'autonomie des nœuds. Elle permet de réduire considérablement le trafic global de la gestion dans le réseau en favorisant les échanges locaux et contribue au passage à l'échelle des infrastructures de gestion.

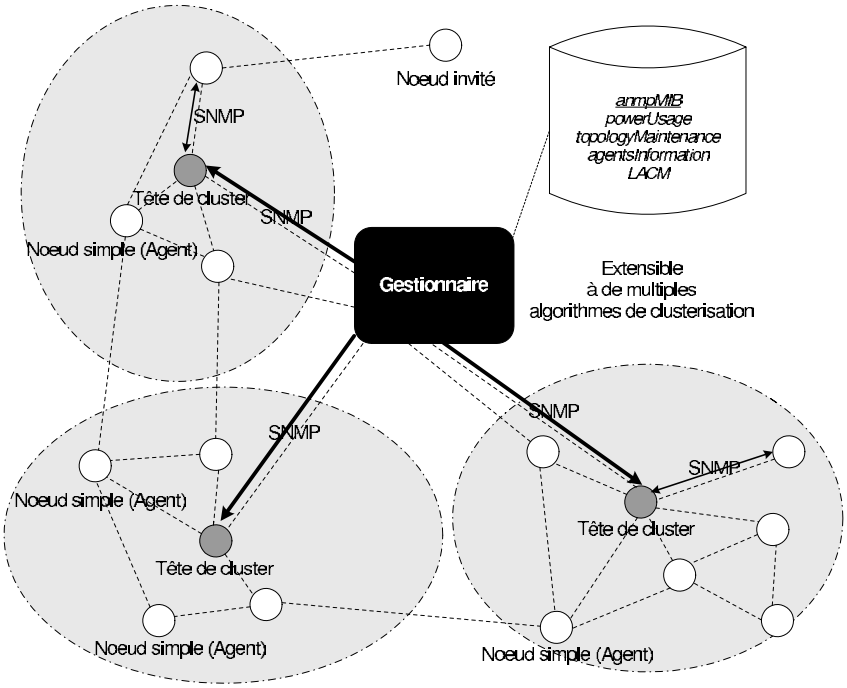


Fig. 2.4 – Architecture ANMP centralisée hiérarchique intégrant la gestion par délégation.

Parmi les approches pionnières pour la gestion des réseaux ad-hoc, l'architecture ANMP (*Ad-Hoc Network Management Protocol*) [53] intègre parfaitement ce mécanisme. Il s'agit d'une architecture centralisée hiérarchique compatible avec le protocole de gestion SNMP, protocole le plus commun dans les infrastructures de gestion actuelles. Cette compatibilité est assurée par l'utilisation du même format de messages de gestion et de la même structure de données. La différence majeure par rapport à la gestion SNMP provient du modèle architectural présenté en figure 2.4 qui combine clusters et niveaux hiérarchiques. Le plan de gestion est décomposé

en clusters et organisé en trois niveaux hiérarchiques comprenant un gestionnaire principal, des gestionnaires locaux qui correspondent aux têtes de clusters et des agents correspondant aux autres nœuds des clusters. La gestion par délégation réduit la charge de trafic en collectant les informations de gestion à des niveaux intermédiaires par le biais des gestionnaires locaux.

L'organisation en clusters est limitée à une hiérarchie à trois niveaux afin de faciliter la maintenance du plan de gestion même dans les scénarios les plus mobiles. ANMP propose deux algorithmes, qui sont définis au niveau applicatif, pour construire ces clusters :

- le premier algorithme est fondé sur les graphes et construit classiquement des clusters de voisins à un saut en fonction de la topologie du réseau. Cette topologie est perçue comme un graphe où chaque entité mobile est représentée par un nœud avec un identifiant unique (adresse MAC, numéro de série par exemple) et un lien bidirectionnel existe entre deux nœuds si et seulement s'ils sont voisins directs. L'algorithme, exécuté de manière distribuée, sélectionne arbitrairement les têtes de clusters qui interviennent comme gestionnaires. Ceux-ci correspondent aux nœuds ayant le plus petit identifiant dans les différents clusters. Chaque nœud implante une structure de données interne qui permet de maintenir la liste des voisins à un saut, la liste des nœuds du cluster courant ainsi que l'identifiant du gestionnaire du cluster.
- le second algorithme repose sur des clusters de voisins jusqu'à trois sauts qui sont définis en fonction de paramètres géographiques et notamment de la densité de nœuds. Les coordonnées des nœuds doivent être fournies par un système de positionnement géographique. A partir de ces données, l'algorithme assure, de manière périodique et centralisée, la construction des clusters en optimisant le nombre de clusters en fonction de la densité du réseau. Un mécanisme distribué permet, entre chaque mise à jour, de gérer la maintenance des clusters lorsque des nœuds se déplacent, en utilisant un protocole d'annonce lorsqu'un nœud quitte ou rejoint un cluster.

L'architecture ANMP ne se limite pas aux deux algorithmes présentés et est extensible à d'autres approches de clusterisation. En particulier, les auteurs d'ANMP établissent une distinction claire entre les clusters au niveau applicatif à des fins de gestion et les clusters au niveau réseau à des fins de routage. Ils mettent cependant en perspective l'intérêt d'exploiter directement un protocole de la couche de routage pour organiser le plan de gestion et ce afin de définir une approche de gestion la plus légère possible.

Une base d'informations de gestion (MIB) est spécifiquement définie pour l'architecture ANMP à travers une extension de la MIB-II [138] standardisée de SNMP. Cette base appelée *anmpMIB* inclut notamment des statistiques sur la consommation d'énergie avec l'utilisation d'un modèle prédictif, des informations de topologie sous la forme de deux sous-groupes pour chacune des approches de clusterisation, et des statistiques sur la prise en charge des agents par les gestionnaires. L'accès à la base d'informations est sécurisé à travers l'implantation du modèle LACM (*Level-based Access Control Model*) [75] avec lequel chaque nœud dispose d'une visibilité différente en fonction de son niveau d'accès.

ANMP définit une architecture centralisée et hiérarchique compatible avec SNMP et qui exploite les mécanismes de délégation. Les algorithmes de clusterisation permettent d'assurer une couverture de près de 90% des nœuds tout en conservant une charge relativement modérée en termes de trafic de gestion. L'utilisation de code mobile est également évoquée afin d'étendre les fonctionnalités des nœuds ad-hoc. Le code mobile est une généralisation de la gestion par délégation qui permet dynamiquement de télécharger et exécuter du code sur les agents.

2.4.2 Gestion par politique

La gestion par politique (PBM, *Policy-Based Management*) [123, 190] est une approche de gestion qui consiste à définir globalement le comportement des équipements du réseau à travers un ensemble de règles génériques du type *si condition alors action* défini par l'administrateur. Ces politiques de haut niveau d'abstraction sont ensuite traduites en commandes de bas niveaux spécifiques aux équipements. Cette approche repose sur l'interaction entre deux composants :

- le point de décision de politiques (PDP, *Policy Decision Point*) permet de prendre en charge les politiques de haut niveau, de s'assurer de leurs cohérences et de les traduire en opérations interprétables par les équipements, pour enfin les distribuer aux points d'application,
- le point d'application de politiques (PEP, *Policy Enforcement Point*) a quant à lui pour rôle l'exécution des opérations de bas niveau transmises par le point de décision.

La distribution de politiques s'appuie sur deux protocoles dédiés COPS [72] et COPS [52] qui correspondent respectivement à deux modèles de distribution différents. Avec le modèle de distribution par externalisation (COPS), le PEP interroge le PDP afin de connaître le comportement à adopter pour tout évènement qui requiert une décision. Avec le modèle par provisionnement (COPS-PR), le PEP dispose d'une base locale de politiques (COPS-PR, *Common Open Policy Service for PRovisioning*) qui est gérée/provisionnée par le PDP. Le PEP peut alors appliquer les politiques contenues dans cette base sans nécessiter l'intervention du PDP. Nous allons présenter trois approches de gestion par politique pour les réseaux ad-hoc : une architecture centralisée appelée DRAMA, un modèle décentralisé du protocole COPS-PR et une distribution de politiques de nœud à nœud.

Gestion par externalisation avec DRAMA

La première approche appelée DRAMA définit une architecture centralisée hiérarchique de gestion par politique présentée dans [50, 49]. Elle est construite en trois niveaux hiérarchiques sous la forme de clusters. Décrite à la figure 2.5, elle est composée d'un ensemble d'agents implantant la même structure et les mêmes fonctionnalités de base. L'agent global (GPA, *Global Policy Agent*) situé au niveau le plus élevé gère un ensemble d'agents de domaine (DPA, *Domain Policy Agent*). Ceux-ci ont eux-mêmes la charge d'agents locaux (LPA, *Local Policy Agent*). Les politiques de gestion sont propagées hiérarchiquement de l'agent global aux agents de domaine, et des agents de domaine aux agents locaux.

Bien que l'architecture soit en tant que telle relativement simple et que les mécanismes de clusterisation ne sont pas clairement spécifiés, un ensemble exhaustif d'expérimentations a pu être réalisé en environnements réels. Divers scénarios d'utilisation ont été évalués à l'aide d'un prototype. Un premier scénario portait sur le monitorage de la consommation CPU des nœuds ad-hoc. Chaque LPA mesure localement sa consommation CPU et transmet les données auprès des DPA. Ceux-ci les aggrègent avant de les envoyer au GPA. Un second scénario avait pour but d'évaluer la capacité de l'architecture à agir sur le réseau à travers des opérations de gestion. Il consistait à réaffecter un serveur de configuration dans le cas de pannes. Chaque agent évalue l'état de fonctionnement du serveur. Si le temps de réponse du serveur courant dépasse une valeur seuil donnée, un nouveau serveur est sélectionné par le GPA. Enfin, un dernier scénario illustrait la gestion par politique de la qualité de service : la répartition de la bande passante pouvait être reconfigurée dynamiquement pour favoriser certaines classes de trafic.

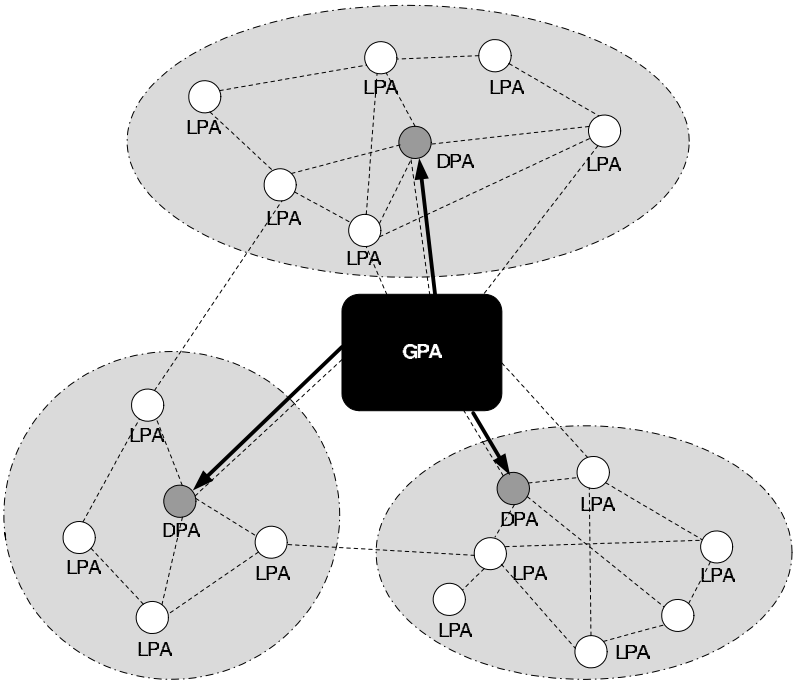


FIG. 2.5 – Modèle centralisé pour la gestion par politique avec l'architecture DRAMA

Décentralisation de la gestion par provisionnement

Une seconde approche introduite par Phanse dans [160, 161] porte sur la décentralisation de la gestion par provisionnement à travers une adaptation du protocole COPS-PR. Il s'agit en fait d'une version hybride de ce protocole qui permet de combiner provisionnement et externalisation à travers des mécanismes de délégation. La figure 2.6 présente cette architecture qui est à la fois distribuée et hiérarchique, et qui intègre notamment trois mécanismes importants :

- **organisation en clusters** : le plan de gestion est organisé en clusters à k-sauts. Durant la phase de déploiement, certains nœuds PDP qualifiés de super-PDP sont initialement présents dans le réseau. Chaque super-PDP est capable de former un cluster avec les nœuds PEP présents dans le voisinage à k-saut. La motivation essentielle étant de borner le temps de réponse du serveur de politiques en limitant la distance entre le serveur super-PDP et les clients PEP.
- **redondance dynamique de services** : ce mécanisme de redondance est utilisé lorsque le nombre de serveurs super-PDP est insuffisant pour servir l'ensemble des nœuds PEP du réseau ad-hoc ou lorsque les clients PEP sont amenés à se déplacer d'un cluster à un autre. Il s'appuie respectivement sur une technique de délégation et une technique de redirection. La délégation est utilisée lorsqu'il manque des serveurs PDP : un super-PDP sélectionne un nœud du réseau et lui délègue des fonctions de gestion afin qu'il intervienne comme PDP. Ce nouveau PDP reste sous le contrôle hiérarchique du super-PDP qui l'a sélectionné. La redirection permet quant à elle de gérer le déplacement des clients PEP de clusters en clusters : le serveur PDP initial est capable d'analyser la distance qui le sépare d'un client PEP donné. Lorsque le client s'éloigne du serveur PDP, celui-ci le redirige vers un nouveau PDP plus proche en assurant la continuité de service.
- **négociation inter-domaines de politiques** : l'objectif est de pouvoir gérer les clients

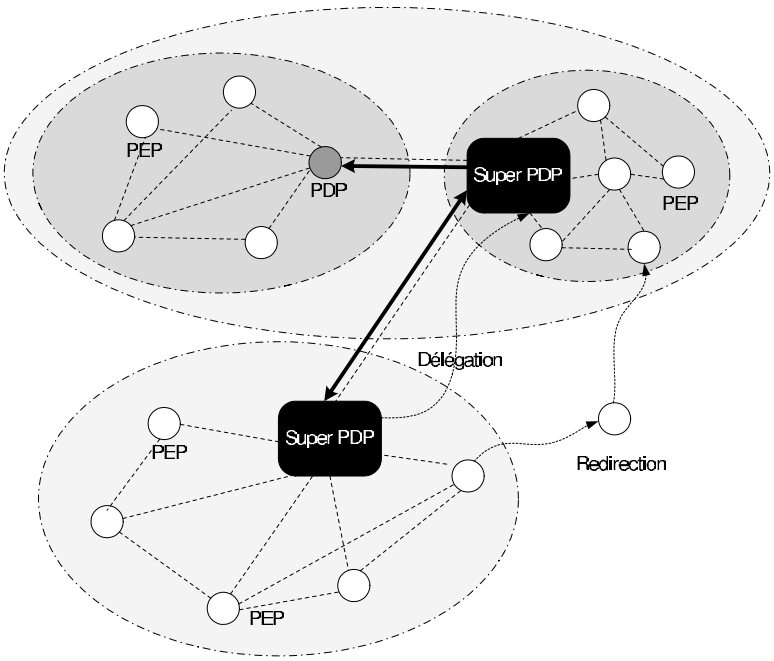


FIG. 2.6 – Modèle décentralisé pour le protocole COPS-PR

PEP provenant d'autres domaines administratifs. Une simple procédure de signalement permet la communication et la négociation de politiques entre serveurs PDP de domaines différents. Lorsqu'un serveur PDP ne peut trouver les politiques pour un client PEP d'un autre domaine, il interroge ce client pour obtenir l'adresse du serveur PDP de son domaine et démarrer une négociation de politiques.

Cette solution hybride entre externalisation et provisionnement a été prototypée et expérimentée en environnements multi-sauts. Cependant, le protocole de communication considéré COPS-PR n'est pas réellement déployé dans les infrastuctures de gestion. Certaines améliorations seraient possibles pour le mécanisme de négociation inter-domaines : les politiques pourraient notamment être transférées entre les serveurs PDP pendant la phase de redirection.

Distribution de politiques de nœud à nœud

Une approche par politique de nœud à nœud est proposée en [145] dans le cadre de réseaux de faible densité. L'objectif est de fournir des niveaux de priorités différents selon les groupes d'utilisateurs en s'appuyant sur le modèle de qualité de service DiffServ [30, 11] qui permet de gérer le trafic réseau par classe de trafic. Dans cette approche de gestion, chaque nœud ad-hoc joue à la fois le rôle de nœud d'accès responsable de la classification et du marquage des paquets et le rôle de nœud central responsable de l'envoi des paquets par niveau de priorité. La distribution de politiques est réalisée de manière complètement décentralisée de nœud à nœud. Lorsqu'un nœud rencontre un nœud voisin, il échange et synchronise ses politiques de gestion avec celui-ci, comme présenté à la figure 2.7.

Pour ce faire, les nœuds ad-hoc implantent à la fois le rôle de serveur PDP et de client PEP tout en maintenant localement une base de politiques. La distribution de politiques s'effectue en deux étapes et intègre un mécanisme de synchronisation :

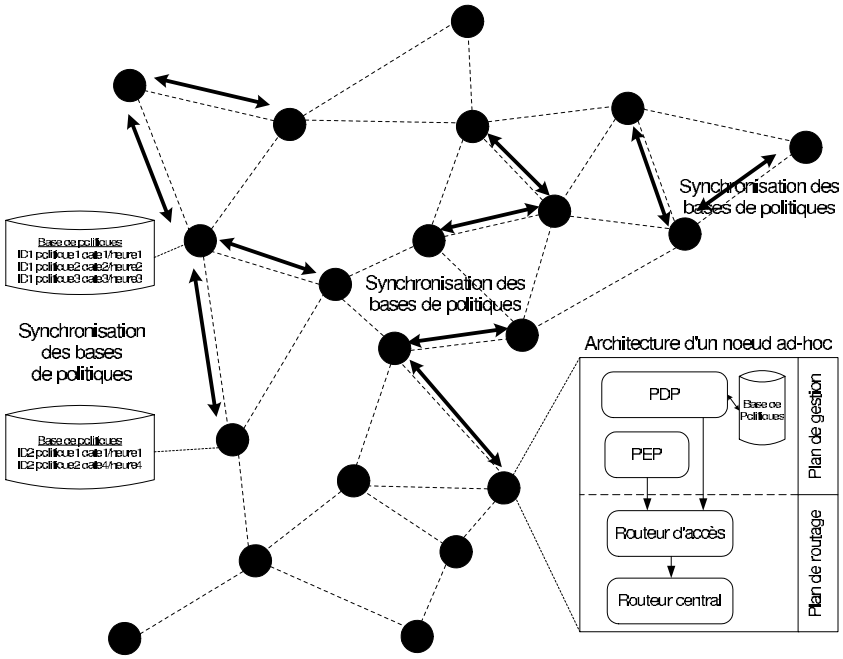


FIG. 2.7 – Propagation de politiques de nœud à nœud

- dans une première étape, l'administrateur définit à partir de sa station ad-hoc les politiques de gestion, en l'occurence les politiques relatives à la qualité de service. Dès lors, le principal problème est de permettre la distribution à l'ensemble des nœuds ad-hoc. La station de l'administrateur intervient comme serveur PDP et distribue les politiques aux clients PEP du réseau. Le protocole de routage permet de joindre l'ensemble des terminaux clients atteignables dans le réseau et de leur transmettre les politiques qui seront stockées dans leur base locale de politiques. Cependant, cette première phase n'est pas suffisante pour configurer le réseau dans son intégralité.
- la seconde étape permet de configurer les nœuds qui n'ont pas pu être configurés directement par le serveur de politiques de l'administrateur car ceux-ci n'étaient pas accessibles à la première étape. Elle consiste à synchroniser les bases de politiques lorsque deux nœuds ad-hoc se rencontrent. Un nœud du réseau, qui a été configuré pendant la première phase, peut intervenir comme PDP afin de configurer un autre nœud voisin qui n'a pas encore reçu la politique.

La synchronisation de politiques s'appuie sur la date de validité des politiques et peut également prendre en compte des niveaux hiérarchiques dans le cas où plusieurs administrateurs gèrent le réseau ad-hoc. Chaque gestionnaire dispose alors d'un niveau d'autorité connu par les nœuds ad-hoc, et il définit des politiques qu'il marque à l'aide d'un identifiant. Les politiques associées au plus haut niveau d'autorité sont toujours priorisées. Cette architecture offre un cadre intéressant pour la configuration d'équipements avec des réseaux ad-hoc de faible taille, en revanche les problèmes de cohérence ne sont pas réellement abordés.

2.4.3 Auto-gestion

La multiplication des infrastructures réseaux et de leurs services a augmenté la complexité des architectures de supervision. La tâche de gestion est devenue de plus en plus complexe pour l'administrateur, ce qui rend l'utilisation de mécanismes d'auto-gestion incontournable [120]. Cette approche de gestion consiste à laisser le réseau prendre en charge sa propre gestion. L'administrateur spécifie les conditions de bon fonctionnement du réseau à travers la définition de politiques. Puis, le réseau choisit et exécute, de manière autonome, les opérations de gestion nécessaires pour atteindre ces objectifs.

L'auto-gestion [146] est une approche qui comprend quatre aires fonctionnelles : l'auto-configuration, l'auto-réparation, l'auto-optimisation et l'auto-protection. L'auto-configuration permet aux équipements de se configurer dynamiquement en fonction des modifications de l'environnement. L'auto-réparation et l'auto-optimisation fournissent aux équipements la capacité de respectivement détecter/réparer leurs propres pannes et d'optimiser leurs propres fonctionnements. Enfin, l'auto-protection joue le rôle de système immunitaire offrant aux systèmes la capacité de se protéger de manière autonome. L'idée consiste systématiquement à offrir au réseau un plus grand pouvoir décisionnel sur lui-même.

La solution d'auto-gestion GUERRILLA proposée dans [177] est conçue spécifiquement pour gérer les réseaux ad-hoc. Elle consiste en une approche contextuelle dans laquelle le pouvoir de décision est distribué entre les nœuds ad-hoc en fonction de leurs capacités. Le but est d'une part de maintenir la connectivité de service dans le plan de gestion en autorisant une gestion déconnectée, et d'autre part de répartir les coûts de gestion en fonction des ressources des nœuds.

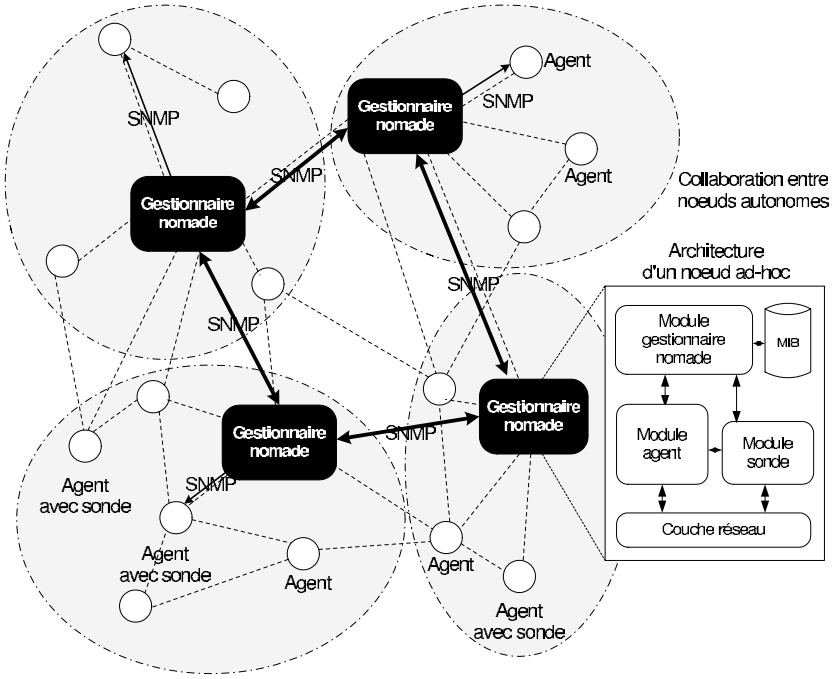


Fig. 2.8 – Auto-gestion avec la plateforme GUERRILLA

Cette architecture distribuée est présentée à la figure 2.8. Le plan de gestion est une fois de plus organisé sous la forme de clusters, mais à l'aide de l'algorithme CLTC (*Cluster-based Topology Control*) [54] qui permet en outre de minimiser la consommation d'énergie. Ce plan

est composé de gestionnaires nomades autonomes ainsi que de sondes actives :

- les gestionnaires nomades maintiennent la connectivité dans le plan de gestion et collaborent ensemble afin de gérer le réseau ad-hoc sans nécessiter l'intervention d'une entité externe. L'attribution du rôle de gestionnaire peut changer en fonction de la topologie, de la densité de nœuds et des ressources disponibles. Si, par exemple un nœud ad-hoc a un niveau de batterie faible, il peut arrêter de jouer le rôle de gestionnaire nomade. Dans ce cas, le pouvoir décisionnel migre dans le réseau en permettant à d'autres nœuds d'intervenir comme gestionnaires. De même, dans le cas où la densité de nœuds augmente, de nouveaux gestionnaires nomades peuvent apparaître dynamiquement afin de répartir la charge de gestion.
- les sondes actives sont déployées par les gestionnaires nomades. L'architecture des sondes actives repose sur une extension d'un agent SNMP qui intègre un environnement d'exécution pour accueillir des scripts spécifiant les opérations de monitorage à réaliser [116, 173]. Les informations de gestion sont collectées et filtrées en fonction des consignes contenues dans le script puis sont transmises au gestionnaire nomade correspondant.

Un mécanisme d'auto-configuration est implanté par les gestionnaires nomades afin de déterminer les opérations de gestion qui doivent être exécutées, et ce de façon adaptative. De manière plus formelle, le gestionnaire nomade dispose d'un ensemble d'opérations de gestion possibles noté A. Il choisit l'opération de gestion à exécuter qui permettra d'améliorer l'état de fonctionnement du réseau. Ce choix correspond à déterminer l'opération a qui maximise la fonction objectif $f(a)$ définie par l'équation 2.1.

$$f(a) = \frac{U(T(s,a)) - U(s)}{C(a)} \; avec \; a \in A \; et \; s \in S \tag{2.1}$$

Dans cette équation, la variable s représente la perception par le gestionnaire de l'état courant du réseau. $U(s)$ défini à partir des politiques spécifie l'utilité [44] du réseau dans son état courant tandis que $C(a)$ représente le coût induit par l'opération de gestion a. La fonction de transition $T(s,a)$ qui peut être associée à chaque action de l'ensemble A permet au gestionnaire d'évaluer l'état futur du réseau en fonction de l'opération de gestion choisie. En maximisant la fonction $f(a)$, le gestionnaire nomade vise à améliorer le fonctionnement du réseau pour un moindre coût. GUERRILLA définit un cadre d'auto-gestion pour les réseaux ad-hoc permettant une gestion davantage adaptative et robuste. Si cette approche permet une parfaite répartition de la tâche de gestion en fonction des capacités et ressources des nœuds, sa mise en œuvre reste plus problématique car elle nécessite une très forte implication de la part des nœuds.

2.4.4 Middleware pour la gestion

Les entités mobiles composant un réseau ad-hoc sont hétérogènes en termes de ressources en énergie, en bande passante et en puissance de calcul. Elles présentent également des disparités importantes en termes de protocoles et de services opérationnels, auxquelles tentent de répondre les solutions à base de middleware [124, 154, 189]. Un middleware programmable pour la gestion des réseaux ad-hoc est proposé dans [97]. Il définit un support homogène pour les communications entre les nœuds ad-hoc, qui est capable de s'adapter dynamiquement aux capacités logicielles de ceux-ci. Ce middleware, fondé sur une architecture centralisée, permet aux nœuds ad-hoc de partager, télécharger et activer des plugins qui offrent diverses fonctionnalités telles qu'un service applicatif ou un protocole de routage. L'objectif est d'élire un ensemble commun de plugins qui satisfait les contraintes diverses des nœuds. Cet ensemble de plugins est distribué aux différents nœuds et constitue un socle homogène de fonctionnalités permettant de faciliter leurs échanges.

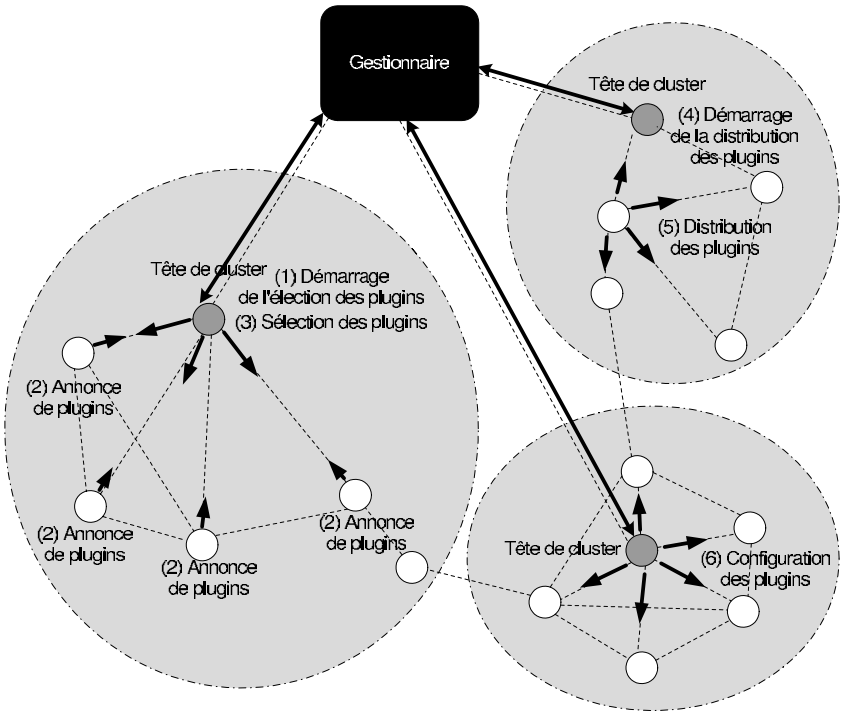


FIG. 2.9 – Middleware pour la gestion

L'architecture du middleware est organisée en clusters. Une tête de cluster est chargée de coordonner l'élection et la distribution des plugins dans un cluster. Chaque nœud ad-hoc d'un cluster donné expose un ou plusieurs plugins dont il dispose du code exécutable dans une base locale. La tête de cluster initialise le mécanisme d'élection en contactant les nœuds du cluster et en leur demandant d'annoncer les plugins candidats. En fonction de l'ensemble de plugins annoncés noté P, elle utilise un algorithme d'élection pour déterminer la base de plugins la plus appropriée pour le contexte. L'élection des plugins a pour but de sélectionner les plugins qui minimisent une fonction de coût notée $g(p)$ décrite par l'équation 2.2.

$$g(p) = \sum_{i=1}^{n} w_i \times C_i(p) \; avec \sum_{i=1}^{n} w_i = 1 \; et \; C_i \in [0, 10], \forall i \in [1, n] \tag{2.2}$$

Les caractéristiques $C_i(p)$ d'un plugin donné p sont affectées d'une note de 0 à 10 en fonction de leur pertinence et sont assignées d'un poids w_i en fonction de la politique de la tête de cluster. La fonction de coût prend en compte les caractéristiques telles que la charge CPU, la consommation de mémoire, la taille du plugin et le système d'exploitation requis, pour converger vers le plus grand ensemble commun de plugins possible. La tête de cluster démarre la distribution de plugins en contactant les nœuds propriétaires afin qu'ils diffusent leurs plugins aux autres nœuds du cluster. Chacun des nœuds sauvegarde les plugins dans une base locale, puis les active. La tête de cluster est capable de configurer et de modifier à la volée les paramètres des plugins courants.

L'implantation du middleware repose sur un protocole léger de messagerie orienté XML et sur la plateforme Java pour petits et moyens équipements. L'évaluation de performances démontre la convergence de l'approche. Néanmoins, le middleware est pour l'instant limité à un unique

cluster. Il devrait être étendu à un scénario qui en intègre plusieurs et qui permet l'établissement de communications inter-clusters.

2.4.5 Evaluation de performances

Un modèle théorique est introduit par Burgess dans [40] afin d'évaluer le passage à l'échelle des approches de gestion pour les réseaux ad-hoc. L'évaluation de performances repose sur un cadre analytique qui prend en compte à la fois la nature non fiable du support physique sans fil et la mobilité des nœuds ad-hoc.

Formellement, le réseau ad-hoc est représenté sous la forme d'une matrice de connectivité notée A où les lignes et colonnes repésentent un nœud du réseau. Une entrée $A[i, j]$ de cette matrice indique que le nœud i a été connecté au nœud j. La connectivité entre deux pairs fluctue dans le temps en fonction du déplacement des nœuds et des perturbations physiques. La valeur $a_{i,j}$ de l'entrée $A[i, j]$ fournit une valeur de probabilité qui indique statistiquement le pourcentage de temps durant lequel les nœuds ont été connectés.

La matrice de connectivité sert de fondement pour déterminer la charge de trafic générée par différentes architectures de gestion par politique. L'analyse du passage à l'échelle repose sur l'axiome suivant : la probabilité d'avoir une configuration opérationnelle dépend de la régularité des opérations de maintenance, qui elles-mêmes dépendent de la fiabilité du support de communication. Afin d'évaluer les performances d'une architecture, le taux moyen d'erreur I_{erreur} généré par un nœud est comparé au taux moyen de réparation $I_{rparation}$. Une configuration est opérationnelle si l'ensemble des erreurs causées par un nœud ont pu être réparées. Ainsi, le taux de réparation $I_{rparation}$ doit rester supérieur au taux d'erreur I_{erreur} afin de limiter le taux de pannes effectives non résolues. Le taux de réparation $I_{rparation}$ dépend de la fiabilité du support physique et de la connectivité entre les nœuds obtenue à partir de la matrice A. Plusieurs modèles organisationnels présentés dans la section 2.2.2 sont décomposés et analysés afin de déterminer le taux de réparation et en déduire le taux de pannes non résolues.

Les résultats de cette analyse fournissent une quantification du passage à l'échelle de différents modèles organisationnels dans le cadre de la gestion par politique. Ils confirment les limites des modèles centralisés causées par le goulot d'étranglement avec le gestionnaire central. Les approches distribuées offrent une meilleure robustesse mais induisent des problèmes de convergence de politiques.

2.5 Synthèse

La gestion de réseaux et de services regroupe l'ensemble des activités qui permet de mettre en œuvre, maintenir et mettre à jour les réseaux et leurs services afin qu'ils répondent aux exigences des usagers spécifiées sous la forme d'objectifs et de contrats de services. Des modèles dédiés permettent d'organiser les infrastructures de supervision de manière homogène et structurée. La dynamique croissante des réseaux et la diversification des services ont contribué à augmenter la complexité de cette tâche.

Les avancées importantes dans les technologies sans fil ont notamment favorisé le développement de réseaux de plus en plus mobiles et décentralisés. Les réseaux ad-hoc en sont une parfaite illustration. Ces réseaux auto-organisés sont formés spontanément à partir d'un ensemble d'entités mobiles communicantes sans nécessiter d'infrastructure fixe pré-existante. S'ils permettent un déploiement rapide et à faible coût, ces réseaux présentent en contre partie des contraintes supplémentaires liées à une topologie dynamique, une bande passante réduite et une durée de vie restreinte due aux limites énergétiques.

De nouveaux modèles et architectures sont requis pour pouvoir intégrer les réseaux ad-hoc dans une démarche de gestion. La topologie dynamique conduit à réorganiser le plan de gestion, typiquement sous la forme de clusters, et à introduire des mécanismes de délégation afin de répartir les activités de gestion. L'absence d'infrastructure fixe favorise les échanges directs de nœud à nœud, par exemple dans le cadre de distribution de politiques, et rend l'utilisation de mécanismes d'auto-gestion incontournable. Enfin, l'hétérogénéité des équipements pousse à établir un socle commun de protocoles et de services en s'appuyant sur des approches à base de middleware. Ces caractéristiques ont également un impact direct sur les domaines d'applications de la gestion.

Chapitre 3

Domaines d'applications

Sommaire

3.1 Introduction

Dans ce chapitre suivant, nous présenterons différents domaines d'applications de la gestion dans le contexte des réseaux ad-hoc. L'objectif est d'illustrer l'intérêt de cette activité à travers des cas concrets. Si de nouveaux modèles et architectures offrent un cadre pour la gestion des réseaux ad-hoc, il est également nécessaire d'évaluer la finalité de cette activité à travers ses applications. Nous analyserons tout d'abord les approches de monitorage qui permettent d'observer le réseau, de mesurer son état de fonctionnement et d'évaluer la consommation de ses ressources. Nous détaillerons ensuite les approches de configuration qui permettent le paramétrage du réseau et de ses entités, en nous intéressant en particulier à la configuration des adresses IP. Enfin, nous décrirons les mécanismes de contrôle qui permettent d'assurer un partage équitable entre les nœuds des ressources et services dont dispose le réseau.

3.2 Monitorage des réseaux ad-hoc

Le monitorage est une activité d'observation qui consiste à évaluer l'état opérationnel et le fonctionnement d'un réseau [55, 24]. Elle permet de déterminer la topologie, l'usage des ressources

ainsi que les performances du réseau en terme de disponibilité et plus généralement en terme de qualité de service. L'activité comprend la mesure, la collecte, l'analyse ainsi que le stockage de données portant sur les paramètres et les mesures de performances réalisées. Les informations collectées par les équipements individuels permettent d'obtenir une vue de plus haut niveau et de produire de la connaissance sur l'environnement complet. Cette connaissance est ensuite utilisée par les équipements eux-mêmes pour réagir de manière intelligente aux changements. Les réseaux ad-hoc amènent à redéfinir de nombreux problèmes tels que : comment déterminer la participation d'un nœud au fonctionnement du réseau, comment stocker et répartir les données collectées, comment synchroniser ces données et optimiser leur durée de vie.

3.2.1 Monitorage local

L'une des premières solutions spécifiquement dédiées au monitorage des réseaux ad-hoc correspond à l'outil WANMON (*Wireless Ad-Hoc Network Monitoring Tool*) [149] qui permet d'analyser localement la consommation des ressources d'un nœud. Un nœud ad-hoc intervient à la fois en tant que terminal et en tant que routeur pour relayer les paquets des autres nœuds dans le cadre de communications multi-sauts. Il est important de pouvoir distinguer clairement les ressources qui ont été consommées pour le fonctionnement du nœud lui-même de celles qui ont été utilisées au service d'autres nœuds.

La solution WANMON est déployée sous la forme d'un agent local qui est responsable du monitorage des ressources. Elle est capable de déterminer statistiquement le coût du routage en termes de trafic réseau, de consommation d'énergie, d'occupation mémoire et de charge CPU. D'une manière plus pratique, si la consommation de la carte réseau peut être obtenue à l'aide d'interfaces dédiées telles que l'interface ACPI (*Advanced Configuration and Power Interface*) [107], elle ne permet pas de distinguer la consommation induite par le trafic réseau routé. Les consommations sont évaluées indirectement à partir des traces réseau obtenues à l'aide d'un analyseur de trafic. Ces traces permettent de calculer la quantité de paquets relayée par le nœud qui, à partir de modèles statistiques, fournissent une approximation de la consommation relative en énergie et en temps processeur. L'architecture WANMON constitue l'un des premiers outils dédiés au monitorage des ressources locales d'un nœud ad-hoc, mais présente un certain nombre de limites : l'outil ne peut pas fonctionner en temps réel car il s'appuie sur une analyse du trafic réseau à posteriori, et il ne s'intéresse qu'à une étude locale du routage à l'échelle d'un nœud.

3.2.2 Monitorage distribué

Afin d'offrir une solution moins locale, DAMON décrit dans [164] une architecture distribuée pour le monitorage des réseaux ad-hoc. Son architecture générique a pour objectif de supporter une grande variété de protocoles, d'équipements et de paramètres. Le système de gestion est composé à la fois d'agents locaux qui monitorent le réseau de manière répartie et de centres de dépôt, également appelés puits, qui permettent de stocker les données de gestion. L'agent est déployé sur une machine hôte, observe le comportement du nœud et transmet les mesures réalisées au centre de dépôt. Le déploiement de l'architecture peut varier en fonction de la taille du réseau. Dans le cas de réseaux de faible taille, le déploiement peut être centralisé avec l'utilisation d'un unique centre de dépôts pour assurer la collecte. Cependant, ce type de déploiement peut induire une forte congestion des routes ainsi qu'une charge excessive du centre de dépôt, dans le cas de réseaux de taille plus importante. DAMON permet alors de répartir la fonction de dépôt à travers l'utilisation de plusieurs puits simultanément. Ceux-ci assurent chacun le support d'un sous-ensemble d'agents et communiquent ensuite entre eux

pour permettre la convergence des données collectées. DAMON définit une solution robuste adaptée à la mobilité des équipements ad-hoc en permettant l'auto-découverte des centres de dépôts par les agents ainsi que la résistance aux pannes de ces centres :

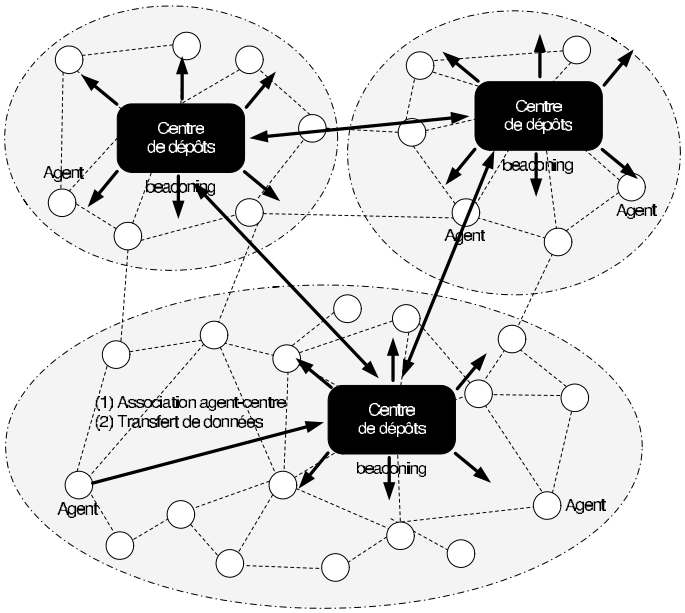


FIG. 3.1 – Monitorage distribué avec DAMON

- l'auto-découverte [58] des centres de dépôts repose sur un mécanisme de *beaconing* [94] périodique qui est initialisé par les dépôts afin d'annoncer leur présence et de diffuser des instructions aux agents. Un agent reçoit ces messages de *beaconing* et maintient une listes des différents dépôts disponibles. Les messages de beaconing sont diffusés dans le réseau et leur propagation est contrôlée par les agents sur la base du nombre de sauts. Ce contrôle permet de limiter la diffusion à un voisinage de nœuds dans le cas de puits multiples. Un agent considère le centre le plus proche comme son centre de dépôt principal et lui transmet les données qu'il a collectées. La distance entre l'agent et le dépôt peut bien évidemment dépasser le voisinage local et être supérieure à un saut. Dans ce cas, les données de plusieurs agents peuvent éventuellement être aggrégées par un agent intermédiaire pour faciliter la collecte.
- la résistance aux pannes [182] des centres de dépôt repose sur une simple opération de permutation du dépôt principal vers un autre dépôt, lorsque le dépôt principal dysfonctionne. L'agent peut détecter la panne d'un centre de dépôts par l'absence de messages de *beaconing* sur une période de temps donnée. Dans ce cas, les agents mettent à jour la liste des centres de dépôts disponibles et envoient les données collectées à un nouveau centre de dépôt. Afin de prévenir une permutation intermittente entre deux dépôts, l'agent ne peut opérer une modification vers un autre centre que s'il reçoit correctement de sa part un nombre donné de messages de *beaconing*.

La solution distingue deux types d'informations de monitorage sur la base de leurs propriétés temporelles et leur assure un traitement différencié : les données fortement dépendantes du temps (informations de topologie) requièrent une mise à jour fréquente et les informations faiblement dépendantes du temps (statistiques journalières sur le trafic réseau) peuvent subir des aggrégations multiples avant d'arriver au centre de dépôts. Pour ce faire, l'architecture d'un agent est

composée d'un module de classification qui permet de répartir les paquets en fonction de leur nature, et d'un module de traitement capable d'aggréger les données faiblement dépendantes du temps.

DAMON définit une architecture distribuée de monitorage dans le contexte des réseaux ad-hoc fondée sur l'utilisation d'un ensemble réparti de centres de dépôts où sont collectées les données de monitorage. Une implantation complète de l'architecture a été réalisée pour observer spécifiquement le comportement du protocole de routage AODV (*Ad-Hoc On-Demand Distance Vector*) [156] et a permis d'évaluer la distribution du trafic de routage pour différents scénarios. Mais, cette architecture reste générique et peut s'appliquer à une multitude d'autres protocoles. Bien que DAMON soit une solution robuste qui peut être facilement maintenue à un moindre coût, le mécanisme d'association fondé sur la proximité pourrait être amélioré afin de permettre une distribution homogène des agents par centre de dépôts.

3.2.3 Synchronisation et réplication

Un réseau ad-hoc étant par nature dynamique et non fiable, des mécanismes de synchronisation sont requis pour maintenir la cohérence temporelle des informations de monitorage. Des techniques de dissémination sont nécessaires pour maximiser la durée de vie de ces informations.

Le premier problème repose sur la synchronisation du temps dans des réseaux distribués et non fiables [180]. Lorsque l'on aggrège des données de monitorage, les relations temporelles (l'événement X a eu lieu avant l'événement Y) et les problèmes de temps réel (les événéments X et Y sont séparés par un certain intervalle de temps) peuvent jouer un rôle important. Le temps logique n'est pas suffisant pour déterminer les relations temporelles car les événements qui apparaissent dans le réseau ne sont pas systématiquement associés à la génération de messages. Il est donc nécessaire de considérer et synchroniser le temps physique des équipements. Cependant, les réseaux ad-hoc peuvent subir de nombreuses partitions et fusions qui affectent les performances des algorithmes de synchronisation classiquement utilisés. Une méthode de synchronisation décrite dans [169] propose d'adapter un algorithme d'estampillage aux conditions des réseaux ad-hoc. Au lieu de chercher à synchroniser les horloges locales des nœuds ad-hoc, l'algorithme s'appuie sur des horloges non synchronisées pour générer des estampilles qui sont ensuite transformées par les nœuds en temps local. La méthode permet de limiter à la fois le nombre de messages échangés ainsi que la consommation en ressources des nœuds ad-hoc.

Le second problème est posé par la perte des données de monitorage. Comme les nœuds ad-hoc ne sont pas fiables, les informations de monitorage maintenues par un nœud local disparaissent du réseau si le nœud tombe en panne ou s'il est déconnecté du réseau. Des techniques de réplication [195, 165, 101] permettent d'améliorer la disponibilité de ces données et d'en augmenter la durée de vie. Elles consistent à copier les informations et de les stocker sur d'autres nœuds du réseau. Il n'est pas possible techniquement d'effectuer une copie sur l'intégralité des nœuds car cette démarche serait bien évidemment trop coûteuse tant en énergie qu'en bande passante. La réplication s'opère usuellement dans le voisinage pour limiter la distance des échanges et préserver l'état des batteries. *A contrario*, l'utilisation de réplicats peut avoir un impact sur l'actualisation des données. Les informations obsolètes doivent être évacuées du réseau. Pour ce faire, il est indispensable d'associer aux informations une durée de vie maximale et d'imposer une mise à jour régulière des données. Une analyse statistique décrite dans [71] étudie les échelles de temps à considérer pour le monitorage des réseaux ad-hoc en faisant apparaître le compromis suivant : la mesure doit s'établir sur une échelle de temps suffisamment courte pour être actuelle mais suffisamment longue pour être significative dans de tels réseaux dynamiques.

3.3 Configuration des réseaux ad-hoc

La configuration des réseaux ad-hoc [103, 170] doit être réalisée de la manière la plus autonome possible afin d'en assurer la robustesse et le passage à l'échelle. Nous constaterons notamment la constitution du groupe de travail MANET Autoconf [133] de l'IETF dédié à l'auto-configuration dans les réseaux ad-hoc. L'un des premiers éléments à configurer est sans aucun doute l'adresse IP des équipements [194, 143], l'objectif étant d'assurer que deux nœuds du réseau ne possèdent jamais la même adresse sur une même période de temps. Cette configuration s'opère dynamiquement dans les réseaux filaires à l'aide du protocole DHCP (*Dynamic Host Configuration Protocol*) [70]. Mais cette solution requiert la présence d'un serveur DHCP central qui maintienne les informations de configuration de l'ensemble des nœuds du réseau. Elle n'est pas adaptée aux réseaux ad-hoc qui doivent pouvoir s'abstraire de toute infrastructure fixe et, autant que possible, de toute administration centralisée. Des nouvelles approches fondées sur un modèle distribué sont nécessaires pour offrir une configuration des adresses IP qui soit robuste et qui puisse résister à de multiples contraintes telles que la mobilité, les pannes, les pertes de messages, les fusions/partitions du réseau et les demandes concurrentes d'adresses. Nous considérons le pire cas, celui d'un réseau ad-hoc complètement autonome qui n'est connecté à aucune infrastructure fixe. Lorsqu'un nœud non configuré souhaite rejoindre le réseau ad-hoc, il faut lui affecter une nouvelle adresse IP. Cette configuration peut être réalisée soit par le nœud lui-même à partir de paramètres internes, soit sous la responsabilité d'autres nœuds qui font déjà partie du réseau. Parmi les solutions proposées, nous pouvons distinguer différentes catégories de configuration [203] : la configuration sans conflit, la configuration avec détection de conflits et la configuration *best effort*.

3.3.1 Configuration sans conflit

La configuration sans conflit consiste à garantir que l'adresse IP d'un nouveau nœud n'a jamais encore été affectée à un autre nœud. Elle s'appuie typiquement sur un paramètre *hardware* unique (numéro de série) pour construire l'adresse IP. L'allocation d'adresse avec le mécanisme d'autoconfiguration sans état d'IPv6 [185] repose sur le simple constat que l'adresse physique d'une carte ethernet est à priori unique, les fabriquants gérant chacun des préfixes différents. L'adresse MAC peut alors être utilisée comme identifiant unique afin de construire l'adresse IP et ainsi éviter les doublons. Par exemple, le nœud ad-hoc peut définir sa propre adresse

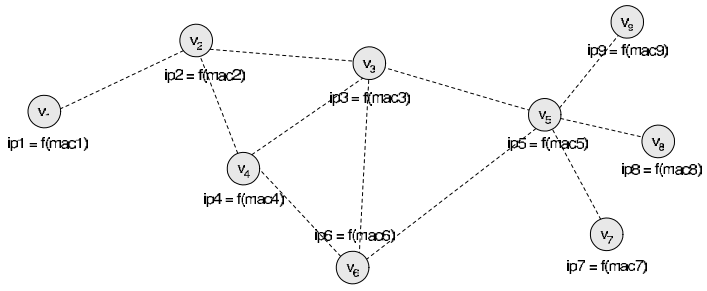


Fig. 3.2 – Configuration sans conflit avec IPv6

IP en combinant le préfixe du réseau et un suffixe qui est calculé localement à partir de son adresse physique. Ainsi, la figure 3.2 illustre un scénario où chaque nœud utilise localement une fonction injective f qui prend en paramètre l'adresse MAC de l'équipement et lui fournit

en retour son adresse IP. Cependant, ce type de solution fondé sur des paramètres *hardware* implique l'unicité des adresses physiques, or celle-ci n'est pas réellement garantie puisqu'il est aujourd'hui facile de reconfigurer les cartes réseaux afin de modifier volontairement l'adresse MAC. L'autoconfiguration sans état IPv6 inclut par conséquent un mécanisme de détection DAD (Duplicate Address Detection)[185] pour gérer les adresses dupliquées.

Un second exemple correspond au protocole DCDP [142] qui utilise des pools d'adresses disjoints pour assurer l'unicité des adresses IP. Ceci suppose que le réseau possède un pool d'adresses disponibles. Le premier nœud du réseau sélectionne une adresse et prend en charge le pool complet. A chaque fois qu'un nouveau nœud rejoint le réseau, il contacte son plus proche voisin. Ce dernier lui affecte une adresse IP et partage son pool d'adresses en deux pools disjoints. Il conserve la première moitié du pool et donne la seconde moitié au nouveau nœud. Un atout de cette approche est qu'elle gère parfaitement bien la partition et la fusion des réseaux. Lorsque le réseau est partitionné, les nœuds des partitions conservent des pools d'adresses différents. Ainsi, aucun doublon ne peut apparaître lorsque deux réseaux fusionnent. Cependant, un des problèmes majeurs de la configuration sans conflit est celle de la réutilisation des adresses. En particulier, si un nœud quitte le réseau sans en faire l'annonce et sans libérer son pool d'adresses, ce dernier devient inutilisable.

3.3.2 Configuration avec détection de conflits

La configuration avec détection de conflits consiste à sélectionner une adresse qui est à priori disponible, puis à vérifier que celle-ci n'a pas encore été allouée en effectuant une requête auprès des autres nœuds du réseau ad-hoc. L'adresse candidate peut être affectée au nouveau nœud, si et seulement si aucun autre nœud du réseau ne possède déjà l'adresse. Dans le cas contraire, le nouveau nœud doit sélectionner une nouvelle adresse candidate et doit répéter la procédure jusqu'à ce qu'il finisse par trouver une adresse disponible.

Dans la solution MANETconf [148], une adresse IP est perçue comme une ressource non partagée d'un système, qui ne peut être utilisée par plusieurs utilisateurs en même temps. L'accès à cette ressource doit donc suivre un algorithme d'exclusion mutuelle. MANETconf s'appuie sur une extension de l'algorithme distribué de Ricart et Agrawala [167], cette extension est capable de supporter un échange non fiable de messages et un nombre dynamique d'utilisateurs.

La demande d'adresses n'est pas réalisée directement par le nouveau nœud mais est formulée par le biais d'un nœud intermédiaire appelé nœud initiateur qui fait déjà partie du réseau ad-hoc. Ceci permet au nouveau nœud de disposer du réseau alors même que l'adresse IP n'a pas été encore affectée, en utilisant le nœud initiateur comme proxy durant l'exécution de l'algorithme. Comme illustré à la figure 3.3, la configuration avec MANETconf se déroule selon les étapes suivantes :

- le nœud demandeur v_1, qui n'est pas configuré, ne dispose pas encore d'une adresse réseau. Il doit sélectionner un nœud initiateur qui pourra lui affecter une adresse IP. Il interroge pour cela son voisinage afin de connaître les nœuds susceptibles d'intervenir comme nœud initiateur et choisit l'un d'entre eux, en l'occurence le nœud v_2.
- le nœud initiateur sélectionne pour le nouveau nœud une adresse réseau qu'il considère comme disponible. Chaque nœud du réseau ad-hoc maintient une liste locale des adresses IP déjà allouées. le nœud initiateur s'appuie sur sa propre liste locale pour choisir l'adresse candidate. Ce choix doit également être validé par les autres nœuds. Le nœud initiateur diffuse alors une requête d'adresse auprès des autres nœuds du réseau.
- à la réception de la requête, chaque nœud du réseau évalue si l'adresse demandée est disponible ou non en analysant sa liste locale des adresses allouées, et envoie en retour une

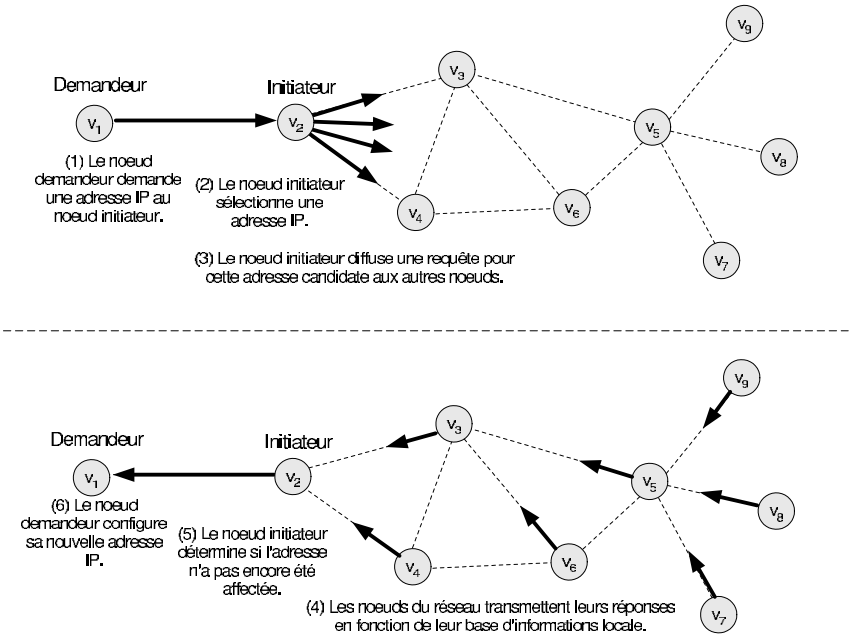


FIG. 3.3 – Configuration avec détection de conflits par MANETconf

réponse respectivement positive ou négative.

– si le nœud initiateur reçoit uniquement des réponses positives alors l'adresse candidate est validée et le nouveau nœud peut être configuré à l'aide de celle-ci. Le nœud initiateur annonce par diffusion aux autres nœuds l'affectation de l'adresse IP et ce afin d'assurer la mise à jour des listes locales. Si en revanche une réponse au moins est négative, cela signifie que l'adresse IP est déjà utilisée par un nœud ad-hoc. Le nœud initiateur doit alors sélectionner une nouvelle adresse candidate et effectuer à nouveau la procédure.

La configuration avec MANETconf repose en fait sur une validation en deux phases [26, 193]. Dans une première phase, les nœuds du réseau valident l'adresse candidate en contactant le nœud initiateur. Dans une seconde phase, le nœud initiateur confirme l'affectation de l'adresse en diffusant une annonce aux nœuds du réseau. Plusieurs demandes d'adresse peuvent être formulées simultanément par différents nœuds initiateurs. Lorsque les adresses demandées sont identiques, les requêtes sont départagées arbitrairement sur la base de l'adresse IP des nœuds initiateurs. Par ailleurs, MANETconf prend en considération divers incidents qui pourraient nuire au bon fonctionnement et à la robustesse de la solution :

– panne d'un nœud : si un nœud quitte le réseau de manière abrupte suite à une panne, il ne peut annoncer son départ aux autres nœuds du réseau. Or cette annonce permet de libérer l'adresse IP du nœud qui quitte le réseau en mettant à jour les listes d'adresses IP déjà allouées. Une technique permet néanmoins de déceler qu'une adresse IP n'est plus utilisée en exploitant les échanges qui ont lieu durant la procédure de configuration. Lors de la requête d'une nouvelle adresse IP, le nœud initiateur reçoit des réponses de la part des différents nœuds du réseau. Aussi, il est possible durant cette étape de détecter de manière indirecte le départ d'un nœud, lorsque celui-ci ne fournit plus de réponses à plusieurs reprises.

– panne d'un nœud initiateur : un nœud initiateur peut tomber en panne avant qu'il n'ait affecté une nouvelle adresse IP au nœud demandeur. Le nœud demandeur maintient un

temporisateur qui lui permet au delà d'un temps donné de resélectionner un nouveau nœud.
- perte de messages : le médium radio n'étant pas fiable, les pertes de messages sont fréquentes dans un réseau ad-hoc. Ceci peut avoir un impact sur la mise à jour des listes d'adresses allouées. Par exemple, si un nœud n'a pas pris connaissance du départ d'un autre nœud, il peut refuser d'affecter une adresse qui est pourtant libérée. Pour éviter ce type d'erreurs, un numéro de séquence est joint à chaque message de mise à jour afin de dater les différentes annonces.
- partition/fusion du réseau : à tout instant, le réseau peut se décomposer en partitions multiples ou peut fusionner avec un autre réseau. Chaque fois qu'une demande d'adresse est émise par un nœud, ce dernier est capable d'évaluer les nœuds présents dans sa partition en fonction des réponses qui lui sont transmises. L'ensemble des adresses allouées dans le réseau correspond à l'ensemble des nœuds qui fait partie de la partition courante. Lorsque deux réseaux ad-hoc fusionnent, les nœuds ad-hoc de partitions différentes échangent les listes d'adresses alouées. Les adresses doublons sont identifiées pour permettre de reconfigurer les nœuds concernés.

MANETconf définit un protocole robuste permettant une configuration décentralisée avec détection de conflits. La configuration est faite indirectement par l'intermédiaire d'un nœud initiateur. Celui-ci peut jouer le rôle de proxy durant l'exécution de l'algorithme afin de permettre au nouveau nœud d'accéder immédiatement au réseau. MANETconf résoud les problèmes de demandes concurrentes d'adresse IP aussi bien que les problèmes liés aux fusions et partitions du réseau. La solution s'inscrit dans le cadre de réseaux autonomes, mais elle pourrait être combinée à un serveur DHCP standard dans le cadre d'architecture hybride filaire/ad-hoc. Bien que MANETconf gère les pertes de messages et la panne de nœuds, les problèmes de sécurité liés aux activités de nœuds malicieux doivent être pris en compte, ce qui n'est pas le cas au moment de la rédaction de ce manuscript.

3.3.3 Configuration *best-effort*

La configuration *best effort* [176] est une approche alternative qui consiste à assigner aux nœuds une nouvelle adresse ayant une très faible probabilité d'être déjà présente dans le réseau. Si elle est très faible, la probabilité d'avoir des doublons reste néanmoins non nulle et une reconfiguration des adresses est nécessaire lorsqu'un conflit d'adresses apparaît.

L'allocation PROPHET [203, 202] définit une telle stratégie en utilisant les propriétés aléatoires d'une fonction à état notée $g(s)$. Cette fonction à état génère des séquences de nombres aléatoires. L'état initial de la fonction $g(s)$ est appelé valeur *seed* et conditionne la séquence de nombres générée par la fonction. Ainsi, différentes valeurs initiales *seed* aboutissent à différentes séquences de nombres. En considérant une adresse IP comme un nombre aléatoire, la fonction à état est utilisée pour fournir des séquences d'adresses. L'objectif est de générer des séquences qui répondent aux deux propriétés suivantes :
- l'intervalle entre deux occurences d'un même nombre dans une même séquence est extrêmement large. On entend par même séquence une séquence générée à partir d'une même valeur initiale *seed*,
- la probabilité d'avoir plus d'une occurence d'un même nombre dans des séquences générées avec des valeurs initiales *seed* différentes est extrêmement faible.

Les propriétés de la fonction permettent d'obtenir des séquences d'adresses différentes telles que la probabilité d'avoir des nombres identiques et donc des adresses dupliquées reste très faible. En s'appuyant sur ces propriétés, la configuration des adresses se déroule de la manière suivante :

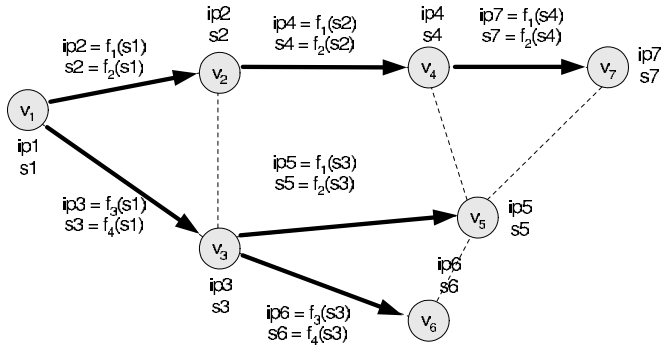


FIG. 3.4 – Configuration *best effort* avec PROPHET

- le premier nœud noté v_1 effectue un tirage aléatoire pour à la fois configurer son adresse IP et choisir la valeur initiale *seed* s_1 pour la fonction à état $g(s)$,
- lorsqu'un nouveau nœud, noté v_2, souhaite rejoindre le réseau, il contacte le nœud v_1 pour obtenir son adresse IP. Le nœud v_1 utilise la fonction à état pour générer l'adresse IP de v_2 mais également pour lui fournir une valeur *seed* initiale s_2,
- le nœud v_2 configure son adresse IP. Il peut également initialiser la fonction à état à l'aide de la valeur *seed* s_2 et peut l'utiliser afin de configurer de nouveaux nœuds.
- les deux nœuds sont désormais capables de générer de nouvelles adresses IP. Quelles que soient les adresses générées par ces nœuds, elles ont une très faible probabilité d'être identiques grâce à la seconde propriété de la fonction.

La procédure est répétée à chaque fois qu'un nouveau nœud souhaite rejoindre le réseau, comme décrit à la figure 3.4. Nous considérons pour cet exemple que les nœuds ont rejoint le réseau par ordre d'indice et que $f_i(s)$ représente la i-ème itération de la fonction à état initialisée avec la valeur *seed* s. Cette même fonction est propagée de nœud en nœud, mais avec des valeurs *seed* initiales différentes. Ainsi les nœuds génèrent des séquences d'adresses disjointes avec un taux extrêmement faible de doublons. PROPHET offre une solution de configuration adaptée aux réseaux de taille importante car il requiert un faible trafic de gestion pour la distribution des adresses. En particulier, la configuration peut être réalisée localement par un nœud voisin, sans nécessiter l'interrogation des autres nœuds du réseau, grâce aux propriétés aléatoires de la fonction à état. Cependant, la solution mise en œuvre ne permet pas de palier totalement la duplication d'adresses et doit également être complétée par un mécanisme de détection des conflits.

3.4 Contrôle des réseaux ad-hoc

Le déploiement d'un réseau ad-hoc est possible grâce à la coopération des nœuds mobiles et à la mise en commun des ressources en bande passante et en énergie. Des mécanismes de contrôle sont nécessaires pour vérifier le bon comportement des nœuds et prendre le cas échéant des mesures correctives [106]. Il s'agit notamment d'assurer un partage équitable des ressources en contrôlant l'accès au médium radio et d'inciter les nœuds à participer aux tâches de fonctionnement du réseau. L'objectif doit être d'une part de récompenser les nœuds qui offrent des services, par exemple ceux qui interviennent comme routeur et relayent le trafic d'autres nœuds, et d'autre part d'identifier les nœuds mal intentionnés ou dysfonctionnants, qui peuvent induire

une surcharge du réseau ou dégrader la qualité de service, afin de limiter leurs impacts sur le réseau.

3.4.1 Contrôle d'accès au médium

Les réseaux ad-hoc représentent un environnement où aucune coordination centrale n'est requise. Le contrôle d'accès au médium, qui permet de partager le support physique, doit pouvoir être opéré de manière complètement distribuée. Le protocole 802.11 [92] largement déployé dans les réseaux sans fil définit deux mécanismes pour la résolution de contention : un mécanisme centralisé PCF (*Point Coordination Function*) [91] qui ne peut être employé que dans des réseaux avec infrastructure car il nécessite un contrôleur central, et un mécanisme distribué DCF (*Distributed Coordination Function*) [90, 29] qui est couramment utilisé aussi bien pour les réseaux avec infrastructures que pour les réseaux ad-hoc.

Le mécanisme distribué DCF utilise la méthode CSMA/CA (*Carrier Sense Multiple Access/-Collision Avoidance*) [45] d'esquive de collision pour résoudre les accès concurrents au médium radio. Un nœud émetteur, qui souhaite transmettre des données, écoute le réseau. Si le canal de communication est encombré, la transmission est différée. Si le canal est libre, le nœud doit encore patienter durant une période de temps *backoff* [100] avant de pouvoir émettre. Cette période de temps est choisie aléatoirement dans un intervalle donné et permet de départager de manière équitable plusieurs nœuds émetteurs. Néanmoins, un nœud mal intentionné peut facilement obtenir un partage inéquitable en sa faveur en ne respectant pas ce protocole [153]. Il lui suffit de volontairement choisir une faible valeur de *backoff* pour que le canal lui soit systématiquement affecté.

Une technique de contrôle définie dans [125] permet d'améliorer le protocole sans fil afin de détecter et gérer les nœuds mal intentionnés. L'objectif est d'offrir au nœud récepteur des fonctions de contrôle qui lui permettent de déceler des comportements abusifs et de pénaliser les nœuds concernés. Le nœud récepteur est capable d'observer les séquences de transmissions des différents nœuds émetteurs. L'analyse d'une courte séquence de transmissions n'est à priori pas suffisante pour distinguer un nœud émetteur qui a légitimement obtenu une faible valeur de *backoff*, d'un nœud malicieux qui n'a pas respecté le protocole. *A contrario*, l'analyse de larges séquences de transmissions permet d'étudier le comportement des nœuds, mais elle est difficilement envisageable du fait de la mobilité des équipements et du délai important requis pour la détection.

La solution proposée offre néanmoins la possibilité de contrôler les nœuds à partir d'observations sur de courtes durées. La sélection aléatoire de la valeur de *backoff* est normalement opérée par l'émetteur. Dans la nouvelle approche, le récepteur prend en charge cette fonction et transmet la valeur dans les messages de contrôle. L'émetteur utilise les valeurs de *backoff* qui lui sont assignées pour les prochaines transmissions avec le récepteur. Grâce à ce transfert de fonctions de l'émetteur au récepteur, le récepteur connaît à l'avance les périodes de *backoff* de l'émetteur. Il peut désormais en de faibles observations identifier si le récepteur respecte ces périodes ou dévie du protocole.

Une déviation du protocole ne signifie pas toujours que le nœud est mal intentionné. S'il a identifié une déviation dans la transmission, le récepteur pénalise dans un premier temps l'émetteur en fonction de la magnitude de la déviation. Pour ce faire, il impose une valeur de *backoff* plus importante pour les prochaines transmissions afin de compenser cette déviation. En revanche, dans un second temps, si la déviation se confirme sur de multiples transmissions, le récepteur considère que l'émetteur est mal intentionné et peut refuser toutes les transmissions initialisées par ce nœud.

Cette approche de contrôle permet d'améliorer le contrôle d'accès au médium radio en facilitant l'identification et la prise en charge de nœuds mal intentionnés. Il s'appuie sur une modification des mécanismes de résolution de contention du protocole et offre un pouvoir plus important au récepteur en lui permettant notamment de choisir les valeurs de *backoff* pour les émetteurs. Cependant, la solution s'intéresse uniquement à une catégorie de comportements abusifs et ne prend pas en compte d'autres scénarios tels que celui de nœuds ad-hoc utilisant plusieurs adresses physiques pour s'accaparer le médium radio.

3.4.2 Contrôle du plan de routage

Le plan de routage peut également être contrôlé afin de déceler le dysfonctionnement de nœuds ad-hoc lors de la tâche de routage [122]. En particulier, la solution *watchdog/pathrater* introduite dans [134] permet d'identifier les nœuds qui ne réussissent pas à transférer des paquets et de limiter leur impact sur le bon fonctionnement du réseau. Ces nœuds, que l'on peut qualifier de *pathologiques*, représentent un problème important en terme de performances car ils peuvent engendrer une dégradation significative du débit moyen offert au sein du réseau. En particulier, une population de nœuds pathologiques d'environ 10% suffit à détériorer le débit moyen de l'ordre de 20%. L'objectif est d'identifier les nœuds pathologiques et d'empêcher par la suite que les paquets à router passent par ces nœuds. L'architecture de contrôle comprend deux composants principaux qui sont déployés sur les différents nœuds :

- le composant *watchdog* ou chien de garde permet d'observer et de détecter les nœuds qui ne fonctionnent pas correctement. Lors d'une communication multi-sauts, les paquets sont routés de nœud intermédiaire en nœud intermédiaire. Nous considérons deux nœuds v_2 et v_3 successifs faisant partie d'un même chemin. Lorsque le nœud v_2 relaie les paquets au nœud v_3, la méthode consiste à ce que le nœud v_2 vérifie si le nœud v_3 continue à relayer correctement les paquets au nœud suivant. Pour ce faire, le nœud v_2 est amené à écouter les transmissions du nœud suivant, en l'occurence le nœud v_3. Le composant *watchdog* maintient en cache les paquets qui ont été envoyés récemment par le nœud v_2 et les compare aux paquets émis par le nœud v_3. Les paquets sont purgés du cache au fur et à mesure des transmissions faites par le nœud suivant. Si un paquet est gardé dans le cache durant un temps supérieur à une valeur seuil, le *watchdog* considère que le nœud intermédiaire n'a pas été capable de relayer le paquet et envoie un message de notification au nœud source du paquet.
- le composant *pathrater* reçoit les messages de notification et maintient un score pour chacun des nœuds du réseau afin d'évaluer leur fiabilité en terme de routage. Lorsqu'un nœud est amené à router de nouveaux paquets, le *pathrater* combine les scores obtenus avec les informations de routage, notamment les informations d'états de liens, afin de déterminer les chemins les plus fiables.

Ce mécanisme de contrôle s'applique au plan de routage et permet de prendre en compte le comportement des nœuds intermédiaires lors de l'acheminement des données. Il permet d'optimiser la construction de routes en sélectionnant des nœuds qui relayent correctement les paquets. L'architecture *watchdog-pathrater* permet de maintenir le débit moyen dans le réseau à un niveau acceptable même dans le cas d'une présence importante de nœuds pathologiques de l'ordre de 40%. Cependant, elle implique que l'ensemble des cartes sans fil dispose du mode promiscuité. Le mode promiscuité permet à un nœud X situé dans le voisinage direct d'un autre nœud Y d'écouter les transmissions de ce nœud, même si elles ne concernent pas directement le nœud X. Il s'agit dans notre cas de pouvoir observer la transmission du nœud v_3 vers le nœud suivant. La solution peut être biaisée par les perturbations physiques et les collisions dans le réseau.

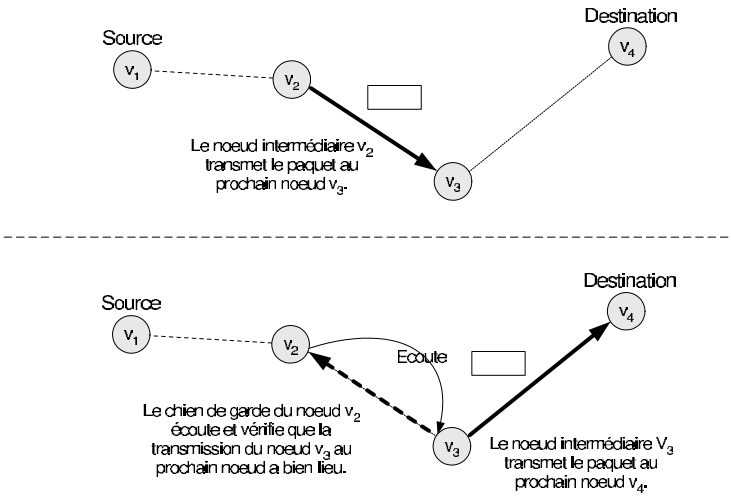


FIG. 3.5 – Modèle *watchdog/pathrater*

Par exemple, Le nœud v_2 peut entendre la transmission du nœud v_3 vers le nœud suivant alors qu'une collision empêche le nœud suivant de recevoir le paquet correctement. Inversement, le nœud suivant peut correctement recevoir le paquet du nœud v_3, alors qu'une collision empêche le nœud v_2 d'entendre cette transmission. Enfin, si les nœuds qui ne relayent pas les paquets correctement n'interviennent plus dans la tâche de routage, l'architecture ne permet pas de les inciter à le faire, ni ne les empêche de continuer à émettre et recevoir des paquets sur le réseau.

3.4.3 Contrôle de services

Une amélioration de la disponibilité de services dans les réseaux ad-hoc peut être obtenue en définissant des modèles économiques tels que proposés dans [42, 43]. Le contrôle consiste d'une part à stimuler la coopération entre les nœuds pour assurer le fonctionnement du réseau, et d'autre part à décourager les nœuds qui tenteraient de surcharger les services.

Comme ces services sont fournis par les nœuds eux-mêmes, leur disponibilité dépend directement de la volonté des nœuds à coopérer. Or la fourniture de services n'est pas directement dans l'intérêt des nœuds, parce qu'elle est consommatrice en bande passante et en énergie. Un nœud est incité de manière naturelle à rester connecté au réseau ad-hoc s'il souhaite pouvoir recevoir de nouveaux messages. En revanche, rien ne l'incite à priori à fournir des services aux autres nœuds. Des mécanismes incitatifs spécifiques permettent de modifier le comportement des nœuds et de les rendre plus favorables à fournir des services [183].

Par ailleurs, les services peuvent devenir indisponibles parce que le réseau est surchargé. Cette surcharge peut provenir aussi bien d'un comportement frauduleux de nœuds qui provoquent une attaque de déni de service, que de nœuds qui souhaitent légitimement utiliser le service mais de manière trop soutenue. Un mécanisme incitatif doit rendre ces comportements coûteux pour les nœuds concernés, afin de les décourager à diffuser abusivement du trafic dans le réseau.

La stimulation de la coopération et la prévention de la congestion peuvent être obtenues conjointement en s'appuyant sur le concept de monnaie virtuelle. L'objectif est de facturer les nœuds qui utilisent les services et de rémunérer ceux qui les fournissent. L'approche [42] introduit une monnaie virtuelle appelée *nugget* qui permet d'améliorer la disponibilité de services. Les nœuds disposent d'un stock initial de *nuggets*. Si un nœud souhaite utiliser un service donné, il

doit payer les autres nœuds pour ce service avec des *nuggets*. D'une part, cela incite les nœuds à fournir des services car c'est le seul moyen pour eux de gagner des *nuggets* et de pouvoir utiliser les services du réseau. D'autre part, cela dissuade les nœuds de surcharger les services inutilement car leur utilisation leur est facturée. L'approche s'intéresse notamment à inciter les nœuds à fournir un service de routage et considère dans ce contexte deux modèles économiques différents : modèle économique de type boursier et modèle économique de type commercial.

Modèle économique de type boursier

Le modèle économique de type boursier [42] consiste à facturer le service au nœud source qui souhaite transmettre les données. Le montant du paiement dépend du nœud destination à atteindre et est réparti entre les nœuds intermédiaires qui servent de relais. Lorsque le nœud source envoie un paquet, il lui adjoint un nombre suffisant de *nuggets* pour atteindre la destination. Durant le routage du paquet, chaque nœud intermédiaire récupère un ou plusieurs *nuggets* pour le service rendu et augmente ainsi son stock de *nuggets*. Le coût en *nuggets* dépend du chemin parcouru par le paquet. Si celui-ci ne présente pas assez de *nuggets* pour atteindre la destination, il sera détruit au cours de son parcours.

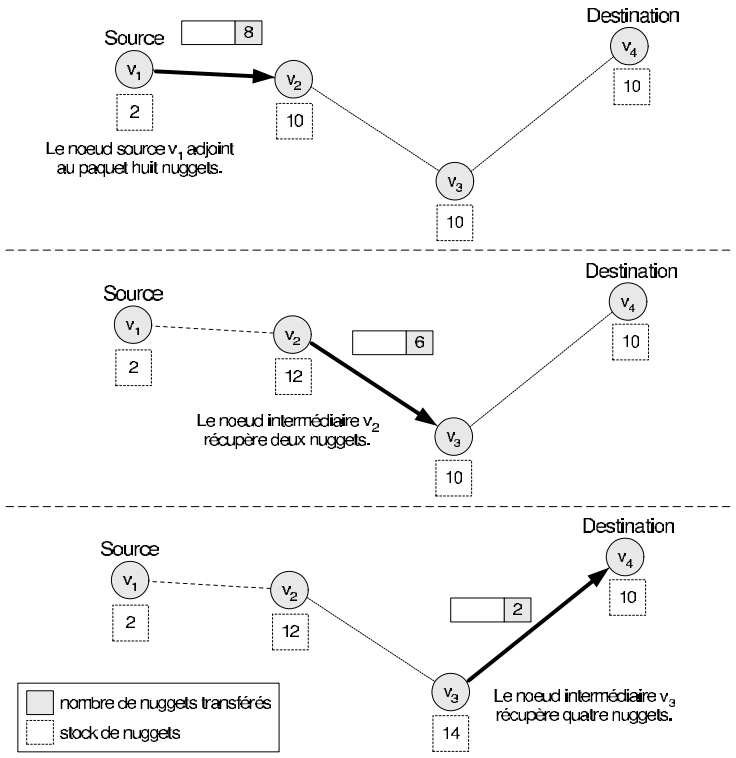


FIG. 3.6 – Modèle économique de type boursier

Un scénario est présenté à la figure 3.6 pour illustrer le fonctionnement de ce modèle. Chaque nœud du réseau dispose initialement de dix *nuggets*. Le nœud source v_1 souhaite transmettre un paquet au nœud destination v_4. Pour ce faire, le nœud v_1 adjoint au paquet huit *nuggets* et l'envoie au prochain nœud v_2. Le nœud v_2 récupère deux *nuggets*, puis relaie le paquet avec les six *nuggets* restants au nœud v_3. Le nœud v_3 prend à son tour quatre *nuggets* puis transmet le paquet au nœud destination v_4 avec les deux *nuggets* restants. Le nombre de *nuggets* facturés est

fixé arbitrairement par les nœuds intermédiaires. Si nous établissons le bilan, nous constatons que les nœuds v_2 et v_3 ont augmenté leur stock de *nuggets* tandis que le nœud v_1 l'a diminué. Le problème élémentaire avec ce modèle est qu'il peut être difficile d'évaluer le nombre de *nuggets* nécessaire pour atteindre une destination donnée. Si le nœud source sous-estime le nombre de *nuggets*, le paquet n'arrivera pas à destination et les *nuggets* auront été dépensés inutilement. Inversement, si le nœud source sur-estime le nombre de *nuggets*, le paquet arrivera à destination mais les *nuggets* inutilisés seront perdus. Une seconde difficulté vient du fait que le paquet lui-même transporte les *nuggets*.

Modèle économique de type commercial

Le modèle économique de type commercial [42] repose sur une négociation financière entre les nœuds intermédiaires. Le paquet ne transporte plus de *nuggets*. Un nœud intermédiaire achète le paquet du nœud précédent pour une certaine somme et le revend au nœud suivant pour une somme plus élevée. De cette façon, chaque nœud intermédiaire facture son service et peut augmenter son stock de *nuggets*. Le coût total est désormais couvert par le nœud destination.

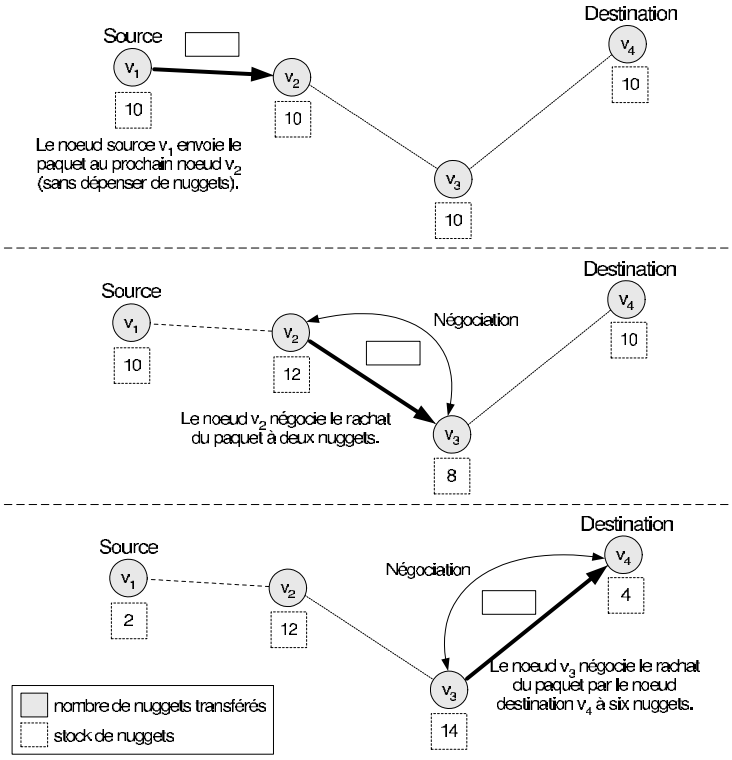


FIG. 3.7 – Modèle économique de type commercial

Le fonctionnement du modèle est décrit à la figure 3.7. Nous considérons le même scénario que précédemment : le nœud v_1 veut transmettre un paquet au nœud v_4. Le nœud v_1 envoie le paquet au premier nœud intermédiaire v_2 pour un coût nul. Le nœud v_2 envoie alors le paquet au nœud suivant pour deux *nugget*. Le nœud v_3 achète le paquet et le revend à la destination finale v_4 pour six *nuggets*. Ainsi, les nœuds intermédiaires v_3 et v_4 ont augmenté respectivement leur stock de deux et quatre *nuggets*. *A contrario*, le nœud destination v_4 a diminué son stock.

L'avantage de ce modèle est que le nœud source n'a pas besoin de connaître à l'avance le

nombre de *nuggets* nécessaire pour acheminer le paquet. Le modèle ne permet cependant pas de dissuader les nœuds de diffuser des paquets inutiles dans le réseau, puisque la facturation est prise en charge par le destinataire. Une solution consiste à permettre aux nœuds intermédiaires de refuser l'achat de paquets lorsque les demandes sont trop fréquentes.

L'approche a pour objectif de stimuler la participation des nœuds afin d'améliorer la disponibilité de services tout en limitant les phénomènes de congestion. Elle repose sur l'utilisation d'une monnaie virtuelle qui permet de rémunérer les nœuds qui participent à l'acheminement des paquets, et ce à travers deux modèles économiques de type boursier et de type commercial. Le premier modèle repose sur une facturation *a priori* à la charge du nœud source tandis que le second modèle repose sur une facturation *a posteriori* à la charge du nœud destination. Une des difficultés de l'approche est d'assurer un provisionnement équitable en *nuggets*. En particulier, un nœud opérationnel qui n'est jamais sollicité par les autres nœuds ne pourra rapidement plus utiliser le service.

3.5 Synthèse

La gestion des réseaux ad-hoc offre de multiples domaines d'applications. L'exploration de ces domaines permet d'identifier en amont la finalité de l'activité de gestion et de déterminer dans quelle mesure elle est conditionnée par les caractéristiques des réseaux ad-hoc. Il s'agit notamment de définir les nouvelles applications en termes de monitorage, de configuration et de contrôle induites par l'auto-organisation du réseau : les nœuds ad-hoc étant à la fois usager et fournisseur de services.

L'activité de monitorage permet d'observer l'état opérationnel du réseau ad-hoc et des éléments qui le composent. Elle permet en particulier de déterminer la part des ressources en bande passante et en énergie consacrée par un nœud donné au fonctionnement du réseau, c'est-à-dire pour le compte des autres nœuds. Les données statistiques du monitorage doivent ensuite être collectées et réparties entre les nœuds ad-hoc de manière optimale, la cohérence des informations pouvant être assurée par des mécanismes de synchronisation et des techniques de réplication permettant d'en augmenter la durée de vie.

L'activité de configuration assure le paramétrage des équipements, avec en premier plan l'adressage des machines. Plusieurs stratégies peuvent être appliquées pour assurer la cohérence de la configuration à l'échelle du réseau ad-hoc : la configuration sans conflit qui garantit que les paramètres des nœuds ne sont jamais incohérents, la configuration avec détection de conflits qui sélectionne les paramètres puis vérifie leurs cohérences en interrogant les autres nœuds du réseau, et la configuration *best effort* qui assigne les paramètres avec une très faible probabilité d'incohérence au sein du réseau.

L'activité de contrôle gère la coopération et le bon comportement des nœuds et prend le cas échéant des mesures correctives. Elle permet le partage équitable des ressources en contrôlant l'accès au médium radio et incite les nœuds à participer au fonctionnement du réseau en récompensant ceux qui offrent des services et *a contrario* en pénalisant ceux qui dysfonctionnent ou ont un comportement mal intentionné.

Les travaux de recherche sur la gestion des réseaux ad-hoc portent donc non seulement sur la conception de nouveaux modèles et architectures, mais également sur l'exploration de ces multiples domaines d'applications.

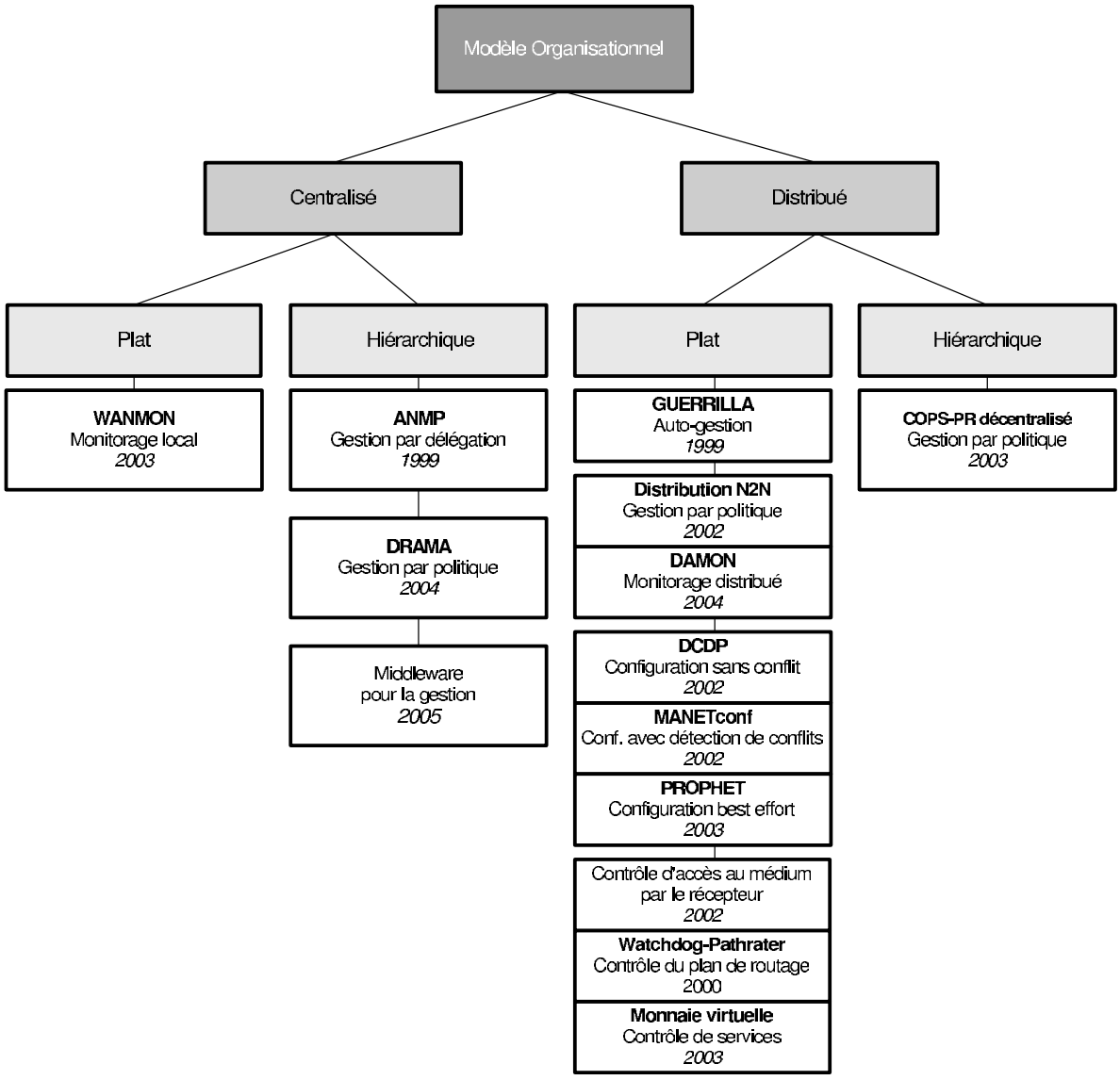


FIG. 3.8 – Récapitulatif des approches par modèle organisationnel

	Fautes	Configuration	Comptabilité	Performances	Sécurité
ANMP Gestion par délégation	✓	✓		✓	✓
DRAMA Gestion par politique		✓		✓	✓
COPS-PR décentralisé Gestion par politique		✓		✓	
Distribution N2N Gestion par politique		✓		✓	
GUERRILLA Auto-gestion		✓		✓	
Middleware pour la gestion		✓			
WANMON Monitorage local		✓		✓	
DAMON Monitorage distribué				✓	
DCDP Conf. sans conflit		✓			
MANETconf Conf. avec détection de conflit		✓			
PROPHET Conf. best effort		✓			
Contrôle d'accès au médium par le récepteur				✓	✓
Watchdog-Pathrater Contrôle du plan de routage	✓			✓	
Monnaie virtuelle Contrôle de services			✓	✓	✓

FIG. 3.9 – Récapitulatif des approches par aire fonctionnelle

Problématique

L'amélioration des technologies mobiles et la standardisation récente des protocoles de routage facilitent aujourd'hui le déploiement et l'utilisation des réseaux ad-hoc. Ces réseaux spontanés permettent de connecter un ensemble d'équipements mobiles indépendamment de toute infrastructure en assurant un acheminement des paquets par sauts successifs de nœud en nœud. La surveillance et le contrôle des réseaux ad-hoc présentent différents verrous tant scientifiques que techniques qui sont à l'origine de nombreux travaux de recherche. Les deux chapitres précédents offrent un aperçu des principaux modèles et architectures dédiés à la supervision de ces réseaux, ainsi qu'une description de leurs différents domaines d'applications. La gestion des réseaux et services ad-hoc est confrontée à de nouveaux défis de recherche qui sont induits par une topologie hautement dynamique et par des ressources fortement contraintes.

Les approches actuelles récapitulées aux figures 3.8 et 3.9 tentent de répondre à ces défis, mais présentent de nombreuses limites théoriques et pratiques. Elles sont tout d'abord souvent mal intégrées à une démarche de supervision globale. Il est indispensable de pouvoir gérer conjointement et de manière homogène des réseaux traditionnels et des réseaux mobiles ad-hoc. Lorsque ces approches de gestion sont intégrées, elles tentent d'appliquer directement des architectures traditionnelles et souffrent en conséquence d'un manque de flexibilité. Il est nécessaire d'offrir une solution véritablement souple qui permette de s'adapter dynamiquement aux changements au sein du réseau ad-hoc. Cette solution devra tirer profit des relations et dépendances qui peuvent s'établir entre les nœuds, les quantifier et les intégrer au cœur des mécanismes de gestion. Enfin, elles restent trop coûteuses en consommation d'énergie et en trafic de gestion. Il est essentiel de s'appuyer sur le plan de routage pour exploiter les informations déjà disponibles, et de limiter l'impact de la supervision par une répartition plus efficace de sa charge entre les nœuds ad-hoc.

La problématique de mes travaux de recherche consiste à établir une nouvelle approche de supervision pour les réseaux et services ad-hoc qui permette de répondre à ce triple besoin d'intégration, de flexiblité et d'économie. Cette approche s'articule autour de trois problèmes de recherche qui portent respectivement sur la construction du modèle d'information, l'organisation du plan de gestion et l'adaptation des opérations de gestion.

Comment étendre le modèle d'information ?

Il est tout d'abord nécessaire de définir une extension du modèle d'information qui permette de spécifier de manière formelle ce qui peut être observé et géré dans un réseau ad-hoc. Les approches de gestion décrites dans les chapitres précédents sont rarement associées à des modèles d'information. Lorsqu'elles le sont, ces modèles d'information sont très partiellement définis. La figure 3.8 détaille les dates de publication de chacune des approches. D'un point de vue chronologique, on constate que la première référence correspond à l'approche ANMP dont la publication a eu lieu en 1999. Les approches sont donc relativement récentes, ce qui peut justifier l'absence de modèle d'information. Le concept des réseaux ad-hoc n'est pourtant

pas nouveau, puisque le premier réseau multi-sauts sans infrastructure a été proposé dans les années soixante-dix. Mais l'engouement pour ces réseaux a conduit à de multiples propositions de protocoles de routage ad-hoc. La standardisation du plan de routage n'a eu lieu que très récemment, ce qui a induit une spécification tardive et partielle du plan de gestion. Notre travail de recherche consiste à identifier les caractéristiques essentielles qui relèvent du contexte ad-hoc afin de structurer l'information de gestion et d'offrir une description homogène des ressources gérées. Nous formaliserons ces informations à l'aide de l'approche standard CIM en intégrant le plan de routage dans cette spécification.

Comment organiser le plan de gestion ?

Il convient ensuite d'établir l'organisation du plan de gestion en s'appuyant sur les relations entre nœuds. Le plan de gestion doit pouvoir s'adapter de manière dynamique aux changements de topologie et permettre une répartition efficace de la charge de gestion afin de minimiser son impact sur la performance des nœuds impliqués. La comparaison des approches par modèle organisationnel à la figure 3.8 révèle la prédominance de deux modèles antagonistes pour la gestion des réseaux ad-hoc. D'une part, le modèle centralisé hiérarchique permet de simplifier les opérations de gestion en ne considérant qu'un seul point de contrôle. Il utilise des nœuds intermédiaires pour réduire la charge de gestion du gestionnaire central et éviter un goulot d'étranglement. D'autre part, le modèle distribué plat permet de fournir de meilleures performances en utilisant un ensemble réparti de gestionnaires, mais implique un degré minimum d'autonomie de la part des nœuds du réseau et peut induire des problèmes de convergence. Indépendamment de ces modèles, notre travail de recherche s'inscrit dans une démarche transverse qui vise à relâcher les contraintes sur le plan de gestion. Au lieu de gérer le réseau dans son intégralité, ne peut-on pas restreindre le plan de gestion à un sous-ensemble de nœuds en fonction de leurs relations ? Cette organisation devra exploiter autant que possible les informations déjà disponibles dans le plan de routage afin de minimiser le coût de la gestion.

Comment adapter les opérations de gestion ?

Notre travail de recherche portera également sur l'adaptation des opérations de gestion aux contraintes des réseaux ad-hoc. Si nous nous appuyons sur la classification FCAPS, nous nous inscrirons à la fois dans le cadre de la gestion de performances et de la gestion de fautes.

En observant la répartition des approches de gestion par aire fonctionnelle récapitulée à la figure 3.9, nous constatons que la gestion de fautes est une aire peu couverte dans le cadre des réseaux ad-hoc. Elle pose pourtant de nouveaux défis liés aux problèmes d'observabilité : la gestion de fautes peut être empêchée par l'impossibilité d'observer un nœud donné. Autant dans un réseau fixe, un nœud qui ne répond pas à un *polling* légitime est habituellement considéré comme non opérationnel. Autant dans un réseau ad-hoc, un nœud peut ne pas être joignable parce qu'il s'est déplacé ou que sa connectivité est temporairement dégradée. Notre objectif consistera donc à évaluer dans quelle mesure nous pouvons rendre cette activité plus robuste en nous appuyant notamment sur la théorie de l'information.

La gestion de performances est pour sa part largement couverte par les approches de gestion présentées à la figure 3.9. Notre approche consistera à analyser les relations entre nœuds afin d'évaluer l'impact d'un nœud ad-hoc sur le fonctionnement du réseau. Cet impact peut être positif (par exemple, les nœuds qui ont une activité de routage importante et constituent une forme de *backbone*) ou négatif (par exemple, les nœuds qui consomment abusivement les ressources du réseau). En particulier, nous déterminerons comment des techniques de filtrage et

de décomposition par vecteur propre peuvent être exploitées afin d'évaluer l'importance relative des nœuds.

Enfin, il est intéressant de remarquer que les travaux de recherche induits par cette problématique peuvent dépasser le cadre des réseaux ad-hoc. En nous plaçant dans un environnement fortement contraint, les solutions mises en œuvre peuvent aboutir à une optimisation des approches de gestion dans le contexte plus général de réseaux et services dynamiques.

Deuxième partie

Une approche de supervision intégrée, flexible et économe pour les réseaux et services ad-hoc

Chapitre 4

Modélisation étendue de l'information de gestion

Sommaire

4.1 Introduction

L'intégration des réseaux et services ad-hoc dans une démarche de supervision passe tout d'abord par une extension du modèle de l'information de gestion. La plupart des approches de gestion actuelle néglige le modèle d'information dans le contexte des réseaux ad-hoc en omettant de le définir ou en le définissant de manière très partielle. Ce modèle de l'information est pourtant essentiel car il définit un cadre formel commun pour la description des ressources gérées et la structuration de l'information de gestion. Il offre une vue homogène et extensible de l'ensemble des ressources et ce quelque soit leur nature, leur localisation et leurs méthodes d'accès. Il permet typiquement de raisonner sur le réseau, de simuler des exécutions d'opérations et d'en fournir une représentation compréhensible et indépendante des spécificités d'implantation des équipements sous-jacents.

Nous présentons dans ce chapitre un modèle d'information de gestion qui permet de décrire ce qui peut être observé et géré dans les réseaux et services ad-hoc. Ce travail de recherche présente un double objectif. D'une part, nous souhaitons identifier les caractéristiques essentielles qui relèvent du contexte ad-hoc. D'autre part, nous souhaitons formaliser ces informations en respectant une approche standard. Nous proposons plus précisément de nous appuyer sur le modèle commun CIM (*Common Information Model*) [56] introduit par le DMTF (*Distributed Management Task Force*) [2]. Ce modèle orienté objet, bien qu'initialement conçu pour la gestion des stations de travail, intègre aujourd'hui de multiples domaines d'applications incluant la gestion de réseaux et de services. Son utilisation dans les infrastructures actuelles s'explique notamment par la simplicité des concepts du langage. Par ailleurs, la richesse des classes disponibles favorise l'extensibilité et la réutilisation du modèle commun.

La standardisation récente des protocoles de routage à destination des réseaux ad-hoc permet d'entrevoir de manière plus concrète leur intégration dans le modèle commun de l'information. Alors que les protocoles de routage tels que BGP (*Border Gateway Protocol*) [200] et OSPF (*Open Shortest Path First*)[144] y sont déjà largement décrits, aucun protocole de routage ad-hoc n'avait encore été introduit au sein de ce modèle. Nous compléterons notre modèle d'information par un sous-modèle dédié au protocole de routage standard ad-hoc OLSR. Ce protocole de routage est une optimisation du routage à état de liens adapté à la nature dynamique des réseaux ad-hoc et constitue l'un des protocoles proactifs les plus répandus dans les réseaux ad-hoc aujourd'hui.

Le chapitre sera organisé en conséquence de la manière suivante : nous présenterons tout d'abord le modèle commun de l'information (CIM) et en rappelerons brièvement les différents concepts. Nous décrirons ensuite notre extension du modèle commun de l'information dédiée à la gestion des réseaux et services ad-hoc. Celle-ci permet de prendre en considération leurs différentes caractéristiques incluant notamment l'organisation du réseau ad-hoc, les échanges entre les différentes entités ainsi que la participation de celles-ci au fonctionnement du réseau. Nous détaillerons enfin dans une troisième section un sous-schéma de notre extension pour gérer les réseaux ad-hoc implantant le protocole de routage standard OLSR.

4.2 Modèle commun de l'information (CIM)

Le modèle commun de l'information (CIM) proposé dans le cadre du consortium WBEM (*Web-Based Enterprise Management*) [135] du DMTF [135]vise à unifier et à étendre les standards d'instrumentation et de gestion pré-existants à travers un formalisme commun pour la spécification des ressources gérées. Ce modèle commun est rapidement devenu un standard incontournable grâce à un soutien fort de la part des industriels, mais aussi grâce à l'utilisation des concepts issus du paradigme orienté objet qui facilite grandement la définition d'une vue conceptuelle des environnements gérés. D'une part, le modèle CIM introduit un formalisme d'expression qui permet de décrire les informations de gestion de manière standard. Les éléments gérés sont organisés sous la forme de diagrammes de classes proches de la spécification UML [65]. Un diagramme (ou schéma) est composé d'un ensemble de classes caractérisées chacune par des attributs et des méthodes, et connectées entre elles à l'aide d'associations. D'autre part, le modèle introduit un ensemble de schémas de référence qui peut être directement réutilisé dans le cadre d'une infrastructure de gestion ou qui peut servir de briques de base pour la construction de nouvelles extensions du modèle commun.

4.2.1 Formalisme d'expression

Le formalisme d'expression précise la manière avec laquelle les informations de gestion doivent être décrites. Il repose sur trois concepts majeurs [81] : un méta-modèle qui spécifie les composants du modèle, un schéma de nommage qui garantit l'unicité des noms des composants et un langage support pour la traduction textuelle du modèle.

Le méta-modèle défini par CIM adopte une approche orientée objet pour la spécification des ressources gérées. Ainsi, si nous considérons comme exemple le diagramme de classes présenté à la figure 4.1, nous retrouvons assez classiquement comme composants de base :

- la classe qui représente une catégorie d'objets gérés. Une classe est caractérisée par des propriétés (ou attributs) et des opérations (ou méthodes). Le méta-modèle distingue trois catégories de classes : les classes concrètes qui peuvent être instanciées à travers un objet géré, les classes abstraites qui définissent un type générique et ne peuvent donc pas être instanciées directement, et enfin les classes d'associations qui représentent chacune une relation particulière entre des classes abstraites ou concrètes. La figure 4.1 présente ainsi un ensemble de cinq classes abstraites parmi lesquelles nous trouvons notamment la classe *ManagedElement* qui définit de manière générique un élément géré.
- l'attribut qui définit une propriété de la classe pour laquelle il est spécifié. Chacun des attributs est caractérisé par un type de données standard (valeur entière, valeur réelle, chaîne de caractères). Ainsi, la classe *ManagedElement* dispose de trois attributs de type chaîne de caractères qui correspondent au nom de la ressource gérée *ElementName* complété par deux descriptions : une description courte *Caption* et une description détaillée *Description*.
- la méthode qui représente une opération que l'on peut exécuter sur un objet géré. Elle est définie par un nom, des paramètres d'entrée/sortie ainsi qu'une valeur de retour. Dans l'exemple présenté à la figure 4.1, la classe *EnabledLogicalElement* dispose d'une méthode *RequestStateChange* qui permet de modifier l'état courant d'un objet géré.
- l'association qui constitue une classe particulière permettant de définir des relations entre classes d'objets gérés et pouvant être complétée par la spécification de cardinalités. A la figure *Caption*, la classe d'association *Component* associe ainsi la classe *ManagedElement* à elle-même pour signifier qu'un élément géré peut être lui-même composé d'un sous-ensemble d'éléments gérés.

Le méta-modèle intègre les mécanismes classiques d'héritage (représentée par une flèche) qui permet à la classe enfant d'hériter des attributs et des méthodes de sa classe parent, ainsi que la notion d'agrégation (représentée par un losange) considérée comme une forme particulière d'association.

Le schéma de nommage introduit un ensemble de conventions pour garantir l'unicité du nom d'un composant. En particulier, le nommage d'un composant s'effectuant dans le contexte d'un schéma, la distinction de composants provenant de schémas différents s'obtient en concaténant le nom du schéma avec celui de la classe. Par ailleurs, des attributs spécifiques appelés clés sont définis dans le schéma de nommage pour permettre l'accès direct à une instance de classe donnée.

La représentation graphique des classes CIM à la figure 4.1 s'apparente très clairement à la modélisation UML. Si cette représentation est intuitive pour un utilisateur, elle doit être complétée par un langage support pour pouvoir être interprétable par un compilateur. Le langage MOF (*Managed Object Format*) permet sa transcription textuelle. Une telle transcription est constituée d'un ensemble de définitions spécifiant les différents composants du modèle.

4.2.2 Schémas de référence

Le modèle CIM définit également un ensemble de schémas de référence dont on distingue le schéma de base (ou *Core Schema*) et le schéma commun (ou *Common Schema*). Ces schémas offrent une structure de référence qui peut être directement réutilisée aussi bien qu'être étendue dans le cadre d'une infrastructure de gestion.

Schéma de base

Le schéma de base [79] définit un ensemble générique de classes et d'associations commun aux différents domaines de gestion. Son objectif est de fournir un socle de base suffisamment général, qui soit composé d'un nombre restreint d'éléments, dont puissent dériver les autres schémas du modèle. Les classes abstraites du schéma de base sont décrites à la figure 4.1. Elles représentent à proprement parler les racines du modèle dont hérite toute autre classe. Hiérarchiquement, la

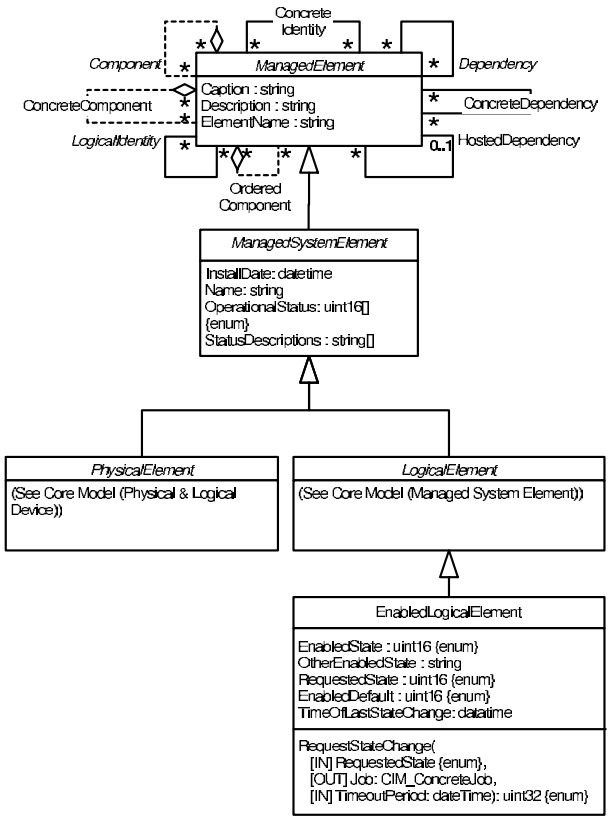


FIG. 4.1 – Diagramme présentant les classes abstraites du schéma de base

première classe correspond à la classe abstraite *ManagedElement* que nous avons déjà décrite précédemment et qui définit de manière très générique un élément géré dans une infrastructure de gestion. La classe *ManagedSystemElement* caractérise pour sa part un élément géré spécifique à un système donné, comme par exemple une station de travail. Le schéma de base distingue ensuite la représentation physique d'un élément *PhysicalElement* (comme par exemple un disque dur) de sa représentation logique *LogicalElement* (comme par exemple un processus). Enfin, la dernière classe *EnabledLogicalElement* représente un élément logique à plusieurs états (actif, inactif, en cours de démarrage, en cours d'arrêt) spécifiés par l'attribut *EnabledState*.

Schéma commun

Le schéma commun définit un ensemble de classes et d'associations spécifiques à des domaines de gestion mais indépendantes de toute implantation particulière. Il hérite du schéma de base et se spécialise en fonction du domaine. Nous pouvons distinguer plusieurs sous-schémas qui permettent de modéliser les applications [80], les réseaux [82], les systèmes [83] en décrivant leurs composants physiques et logiques. Notre intérêt porte tout particulièrement sur le sous-schéma relatif aux réseaux qui a permis d'ouvrir le modèle d'information CIM à la gestion de réseaux. Il est aujourd'hui complété par un sous-schéma prenant en charge la gestion par politique.

En complément de ces schémas, les **schémas d'extension** permettent de définir de nouvelles classes afin d'étendre ou de spécialiser le modèle d'information à d'autres domaines de gestion qui ne sont pas encore couverts par le schéma commun.

4.3 Schéma d'extension pour les réseaux et services ad-hoc

Nous avons défini un schéma d'extension permettant d'intégrer les réseaux et services ad-hoc au sein du modèle commun CIM. L'objectif consiste à proposer un modèle d'information suffisamment générique pour être applicable à toute forme de réseaux ad-hoc et ainsi faciliter la prise en charge de ceux-ci au sein d'une infrastructure de supervision. De part sa nature, ce schéma d'extension exploite à la fois les classes et associations du schéma de base mais aussi un sous-ensemble de classes issues du schéma commun. La figure 4.2 offre un aperçu de notre

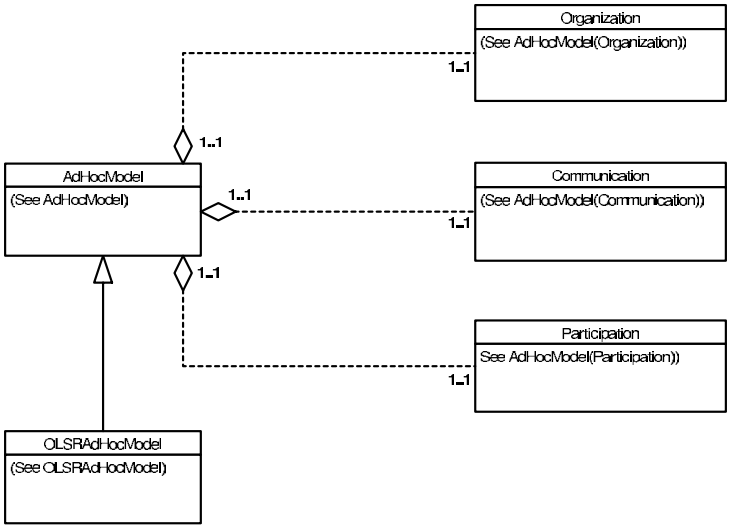


FIG. 4.2 – Aperçu du schéma d'extension pour les réseaux et services ad-hoc

modèle d'information. Nous avons identifié les éléments caractéristiques d'un réseau ad-hoc et les avons formalisés et regroupés en trois sous-schémas principaux portant respectivement sur l'organisation, la communication et la participation au sein du réseau. Nous introduirons par la suite un quatrième sous-schéma pour spécialiser notre modèle d'information afin de prendre en considération les spécificités des réseaux ad-hoc implantant le protocole de routage standard OLSR.

4.3.1 Organisation

Le premier sous-schéma représenté à la figure 4.3 porte sur l'organisation du réseau ad-hoc. La partie grisée correspond aux classes et relations du sous-schéma, tandis que la partie non grisée identifie les classes originelles provenant du schéma de base et du schéma commun qui servent de support à la construction de notre modèle. L'objectif vise à définir une organisation du réseau qui soit suffisamment générique pour s'appliquer aux différentes architectures de supervision présentées précédemment.

Nœud ad-hoc

L'élément de base correspond au nœud ad-hoc représenté par la classe *AdHocNode*. Celle-ci hérite de la classe *System* définie dans le schéma commun. Le nœud est identifié à l'aide d'un identifiant unique représenté par l'attribut *NodeID*. Il est également caractérisé par les attributs *Name* et *CreationClassName* obtenus par héritage de la classe *System*. L'attribut *CreationClassName* que l'on retrouve pour les trois classes du sous-schéma indique la classe effective qui a permis l'instanciation de l'objet. Plus précisément, cet attribut permet au modèle CIM de distinguer plusieurs objets disposant d'attributs similaires mais issus de sous-classes différentes qui héritent de la même classe.

Du fait de sa mobilité et de ses contraintes en énergie et en bande passante, le nœud est susceptible de quitter ou de rejoindre le réseau à tout instant. L'attribut *JoinTime* indique la date à laquelle le nœud à rejoint le réseau, tandis que l'attribut *LeaveTime* précise la date à laquelle il a quitté le réseau pour la dernière fois. L'attribut *LeaveTime* peut facilement être renseigné si le nœud quitte proprement le réseau, c'est-à-dire si un protocole d'annonce lui permet d'informer les autres nœuds de son départ. En revanche, si le nœud quitte le réseau de manière abrupte, le départ du nœud est uniquement pris en compte lorsque celui-ci n'a pas donné présence de vie au delà d'une certaine période de temps. Un dernier attribut *PredictiveLifeTime* permet optionnellement de fournir une information sur la durée de vie potentielle du nœud en s'appuyant notamment sur l'état énergétique de la batterie [131], et ce afin de limiter l'importance du nœud au sein du réseau (comme par exemple, ne plus lui affecter le rôle de gestionnaire) lorsque son départ est imminent.

Cluster de nœuds

Les nœuds ad-hoc sont ensuite regroupés sous la forme de clusters. La manière habituelle d'organiser le réseau aurait été de considérer une hiérarchie de sous-réseaux, mais cette organisation est trop rigide pour des environnements aussi dynamiques, bien qu'il soit possible de définir une configuration des adresses IP avec préfixe tel que proposé dans [114]. Un cluster de nœuds ad-hoc est représenté par la classe *ClusterOfAdHocNodes*. Elle dispose de l'attribut *ClusterId* complété par les attributs *Name* et *ClassCreationName* obtenus par héritage. Le nombre de nœuds ad-hoc présent dans le cluster est également spécifié, à l'aide de l'attribut *NumberOfNodes*.

La classe*ClusterOfAdHocNodes* hérite de la classe *AdminDomain* qui est définie dans le schéma commun et représente un ensemble générique d'éléments appartenant à un même domaine administratif. La classe est reliée à la classe *AdHocNode* par une relation d'agrégation définie par la classe *ClusterMates*. La cardinalité indique qu'un cluster est au moins composé d'un nœud ad-hoc mais qu'un nœud peut appartenir à plusieurs clusters. Un cluster est sous la responsabilité d'un ou plusieurs gestionnaires locaux tels que spécifiés par la classe d'association *LocalManagers*. La vie d'un cluster de nœuds est éphèmère : elle est caractérisée par une date

de déploiement (attribut *DeploymentTime*), une date de fin de vie (attribut *EndLifeTime*) et optionnellement une évaluation de la durée de vie (attribut *PredictiveLifeTime*).

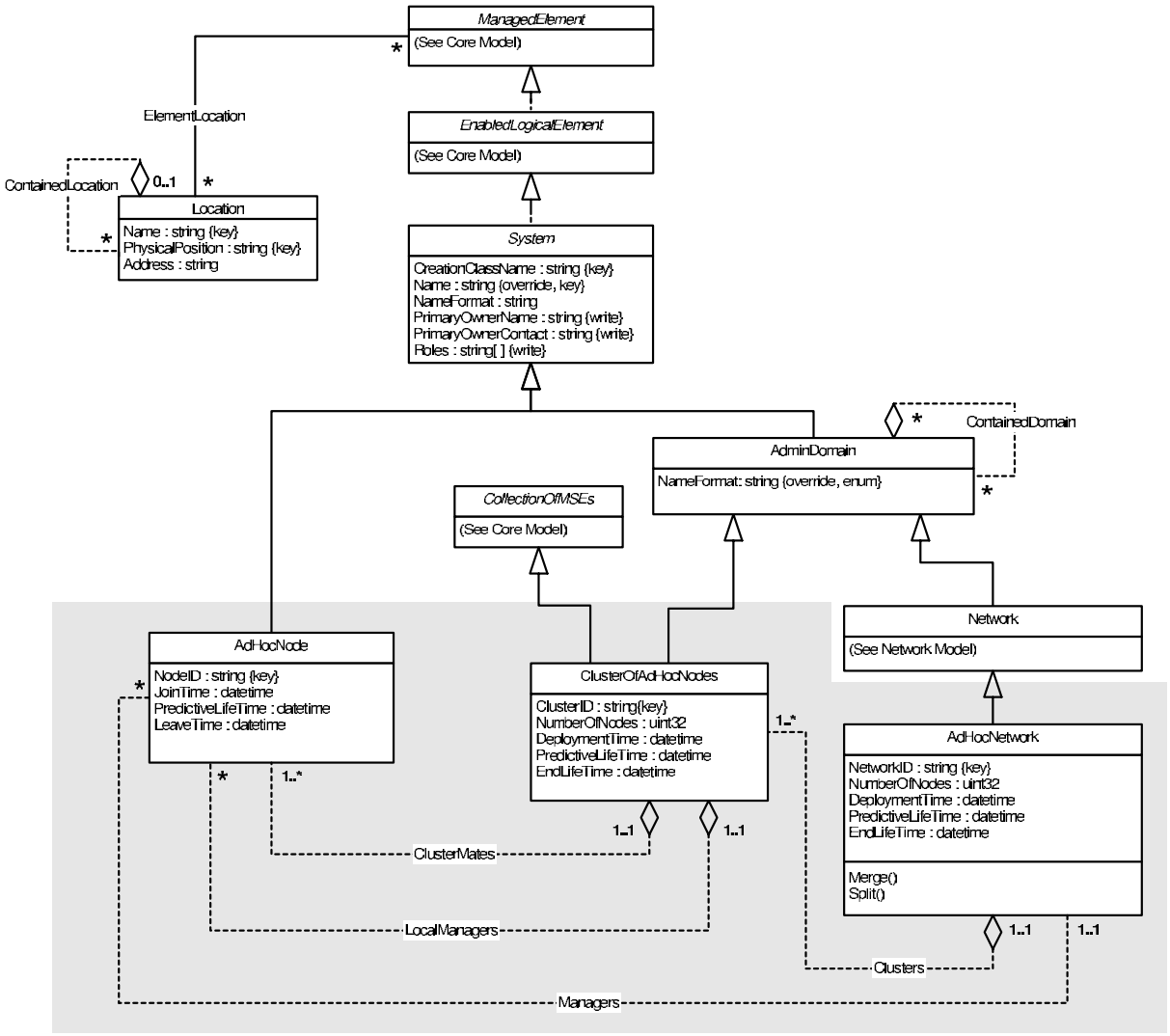


FIG. 4.3 – Diagramme de classes du sous-schéma *Organization*

Réseau ad-hoc

Un réseau ad-hoc est organisé comme un ensemble de clusters de nœuds. Il est représenté par la classe *AdHocNetwork* dans notre modèle d'information. La classe hérite de la classe *Network* définie dans le schéma commun, elle-même définissant un domaine administratif à travers l'héritage de la classe *AdminDomain*. Elle se caractérise par différents attributs permettant de l'identifier et de définir le ou les nœuds gestionnaires du réseau à l'aide de la classe d'association *Managers*. La relation de contenance réseau/cluster est également spécifiée à travers la classe d'association *Clusters* reliant la classe *ClusterOfNodes* à la classe *AdHocNetwork*. Le déploiement du réseau est effectué de manière spontanée si bien que nous retrouvons des propriétés temporelles spécifiées à l'aide des attributs *DeploymentTime*, *PredictiveLifeTime* et *EndLifeTime*. La mobilité des nœuds implique que le réseau ad-hoc peut être partitionné en plusieurs réseaux de

taille plus réduite, ou peut fusionner avec un autre réseau pour former un réseau de taille plus importante. Les méthodes *Split()* et *Merge()* de la classe *AdHocNetwork* permettent de prendre compte les fusions et partitions au niveau du modèle d'information.

Nous avons détaillé les propriétés temporelles des trois classes *AdHocNetwork*, *ClusterOfAdHocNodes* et *AdHocNode*. Leurs propriétés spatiales permettant de rendre compte de la topologie du réseau ne nécessitent pas d'être introduites spécifiquement. Elles sont directement héritées de la classe *ManagedElement* qui est associée à la classe *Location* fournissant la position géographique de l'élément géré.

4.3.2 Communication

L'objectif d'un réseau ad-hoc est de permettre les échanges entre nœuds ad-hoc à travers l'établissement de communications directes ou de communications multi-sauts. Lorsqu'un nœud ad-hoc source ne se situe pas dans le voisinage direct du nœud destination, des nœuds intermédiaires interviennent en tant que routeurs pour transmettre les messages. Le second sous-schéma porte en l'occurrence sur les communications au sein d'un réseau ad-hoc à différentes échelles. Le diagramme de classes correspondant est présenté à la figure 4.4. Les communications doivent pouvoir être inventoriées et quantifiées au niveau microscopique : un nœud communiquant avec d'autres nœuds, mais également au niveau macroscopique à l'échelle d'un cluster ou à l'échelle du réseau tout entier : peut-on distinguer des sous-ensembles de nœuds présentant des échanges importants ? Comment comparer l'activité d'un réseau ad-hoc par rapport à celle d'un autre réseau ?

Le schéma commun définit la classe abstraite *StatisticalInformation* pour décrire les informations statistiques portant sur un élément géré. La classe abstraite est ensuite spécialisée en fonction de la nature de l'élément à travers des sous-classes. Typiquement, la sous-classe *SystemStatisticalInformation* représente les informations statistiques pour un système. Elle est associée à la classe *System* à l'aide de la classe d'association *SystemStatistics*. Nous avons souhaité spécialiser les informations statistiques relatives aux échanges IP (classe *IPstatisticalInformation*) et aux échanges niveau MAC (classe *MACStatisticalInformation*, non représentée sur la figure) dans le cadre des trois classes précédemment définies (*AdHocNetwork*, *ClusterOfAdHocNodes*, *AdHocNode*) correspondant respectivement à trois niveaux d'échelle différents.

Echanges niveau IP

La classe *IPstatisticalInformation* renseigne sur le trafic réseau tel qu'il est possible de l'analyser à partir d'une interface IP [111]. Nous y retrouvons le nombre de paquets reçus (attribut *ReceivedPDUs*), le nombre de paquets émis (attribut *SentPDUs*), le nombre de paquets délivrés (attribut *PDUDelivers*), le nombre de paquets retransmis (attribut *ReceivedPDUForwards*), ainsi que des attributs additionnels portant sur les paquets erronés et rejetés.

Nous nous focalisons dans un premier temps sur les échanges à l'échelle d'un nœud ad-hoc en introduisant deux classes *AdHocNodeIPStatistics* et *InterAdHocNodesIPStatistics* héritant de la classe *IPStatisticalInformation*. La classe *AdHocNodeIPStatistics* indique les statistiques réseau pour un nœud donné en agrégeant le nombre de paquets échangés sur les différentes interfaces IP. Afin de faire apparaître les dépendances entre nœuds, la seconde classe *InterAdHocNodesIPStatistics* détaille les échanges établis entre deux nœuds ad-hoc donnés. Les classes d'association *NodeA* et *NodeB* permettent d'associer deux nœuds à cette classe de statistiques. Les attributs *FromAtoB* et *FromBtoA* de type booléen définissent le sens des échanges, respectivement du nœud A vers le nœud B et du nœud B vers le nœud A.

A un niveau intermédiaire, les échanges IP d'un cluster sont représentés par les classes *AdHocClusterIPStatistics* et *InterAdHocClustersIPStatistics*. Les statistiques définies dans la première classe sont obtenues en agrégeant les statistiques niveau IP de l'ensemble des nœuds ad-hoc associés au cluster considéré. Par ailleurs, la classe *InterAdHocClustersIPStatistics* indique

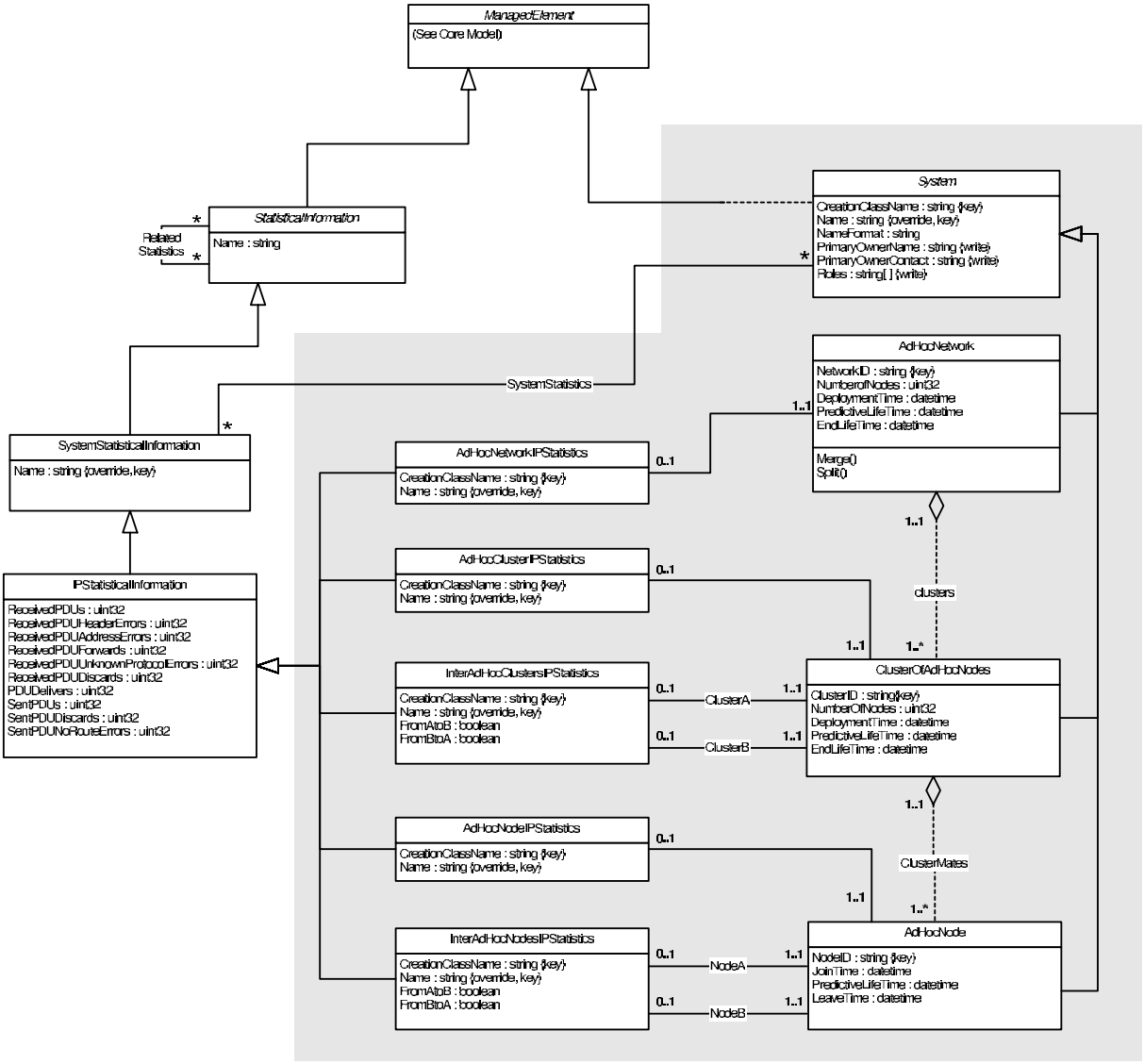


FIG. 4.4 – Diagramme de classes du sous-schéma *Communication*

les échanges entre deux clusters donnés, représentés respectivement par les classes d'association *ClusterA* et *ClusterB*. Conceptuellement, un cluster A échange avec un cluster B lorsqu'un nœud quelconque du cluster A échange avec un nœud quelconque du cluster B. Le sens des échanges est spécifié à l'aide des attributs *FromAtoB* et *FromBtoA*.

Enfin, la classe *AdHocNetworkIPStatistics* représente les statistiques IP à l'échelle du réseau ad-hoc. Les statistiques peuvent être obtenues en agrégeant les valeurs de chacun des nœuds ad-hoc du réseau. L'agrégation des statistiques provenant des différents clusters est également envisageable à la condition stricte que la configuration considérée ne permette pas à un nœud ad-hoc d'appartenir à plus d'un cluster, sous peine d'obtenir des statistiques biaisées par une

sur-évaluation.

Echanges niveau MAC

Parallèlement, la classe *MACstatisticalInformation* (non réprésentée sur le diagramme) renseigne sur les communications établies au niveau de la couche d'accès au médium [92]. Nous avons dérivé cinq classes *AdHocNetworkMACStatistics*, *AdHocClusterMACStatistics*, *InterAdHocClustersMACStatistics*, *AdHocNodeMACStatistics*s et *InterAdHocNodesMACStatistics* par héritage de la classe *MACstatisticalInformation*. Elles permettent de manière similaire de détailler les échanges niveau MAC à l'échelle du nœud, à l'échelle du cluster et à l'échelle du réseau. On remarquera en particulier que la classe *AdHocNodeMACStatistics* permet classiquement d'évaluer l'état de liens du nœud courant avec son voisinage direct.

Par ailleurs, les échanges niveau MAC à une échelle plus large (cluster ou réseau) rendent compte de l'activité des nœuds situés en bordure. Conceptuellement, un cluster A échange avec un cluster B au niveau MAC lorsqu'une trame contient comme adresse source l'adresse MAC d'un nœud du cluster A et comme adresse destination l'adresse MAC d'un nœud du cluster B (ou inversement). Par conséquent, un tel échange implique l'existence d'une liaison directe entre deux nœuds de clusters différents. Les nœuds considérés sont donc les nœuds situés en bordure de clusters, c'est à dire typiquement à un saut du nœud d'un autre cluster.

4.3.3 Participation

Un réseau ad-hoc est un réseau auto-organisé dont le fonctionnement repose sur la participation des nœuds [139]. Un nœud ad-hoc intervient à la fois en tant que terminal pour son propre compte et en tant que routeur pour les autres nœuds dans le contexte de communications multi-sauts. En particulier, l'outil WANMON [149] permet de déterminer statistiquement le coût de l'activité de routage en terme de trafic réseau, mais aussi en termes de consommation d'énergie, de charge CPU et d'occupation mémoire. Nous souhaitions définir un cadre suffisamment générique qui permette d'évaluer l'implication d'un nœud ad-hoc dans le fonctionnement du réseau.

Le sous-schéma *Participation* de notre modèle d'information étend les classes du schéma commun pour spécifier la participation d'un nœud ad-hoc en tenant compte de sa dualité routeur/terminal. Nous nous sommes appuyés sur la classe abstraite *StatiscalInformation* dont nous avons dérivé la classe *AdHocParticipationStatisticalInformation*. Cette dernière indique la participation d'un nœud ad-hoc sous la forme d'un attribut *ParticipationRatio* qui de manière conceptuelle rend compte de l'activité d'un nœud en tant que routeur relativement à son activité en tant que terminal. Aussi, la classe *AdHocParticipationStatisticalInformation* est associée à la classe *StatisticalInformation* à travers deux classes d'association nommées *Terminal* et *Router* qui traduisent des informations statistiques de même nature, mais en distinguant respectivement celles portant sur l'activité du nœud en tant que terminal et celles portant sur son activité en tant que routeur.

La classe *StatisticalInformation* se décline en plusieurs classes concrètes spécifiques à un domaine particulier. De manière similaire, nous proposons de décliner la classe *AdHocParticipationStatisticalInformation* pour être appliquée sur les différentes formes d'éléments gérés. Ainsi, la classe *AdHocParticipationPhysicalStatInformation* renseigne sur la participation du nœud en termes de ressources physiques. Elle hérite de la classe *AdHocParticipationStatisticalInformation* et est reliée à la classe *PhysicalStatisticalInformation*, elle-même associée à la classe *PhysicalElement*. La classe *AdHocParticipationStatisticalInformation* est également dérivée pour

les éléments logiques de l'environnement géré. En particulier, la classe *AdHocParticipationServiceStatInformation* peut être utilisée pour traduire l'exploitation du service de routage. Elle est associée à la classe *ServiceStatisticalInformation*, elle-même liée à la classe *Service* par la classe d'association *ServiceStatistics*. Ainsi, la participation d'un nœud au fonctionnement d'un

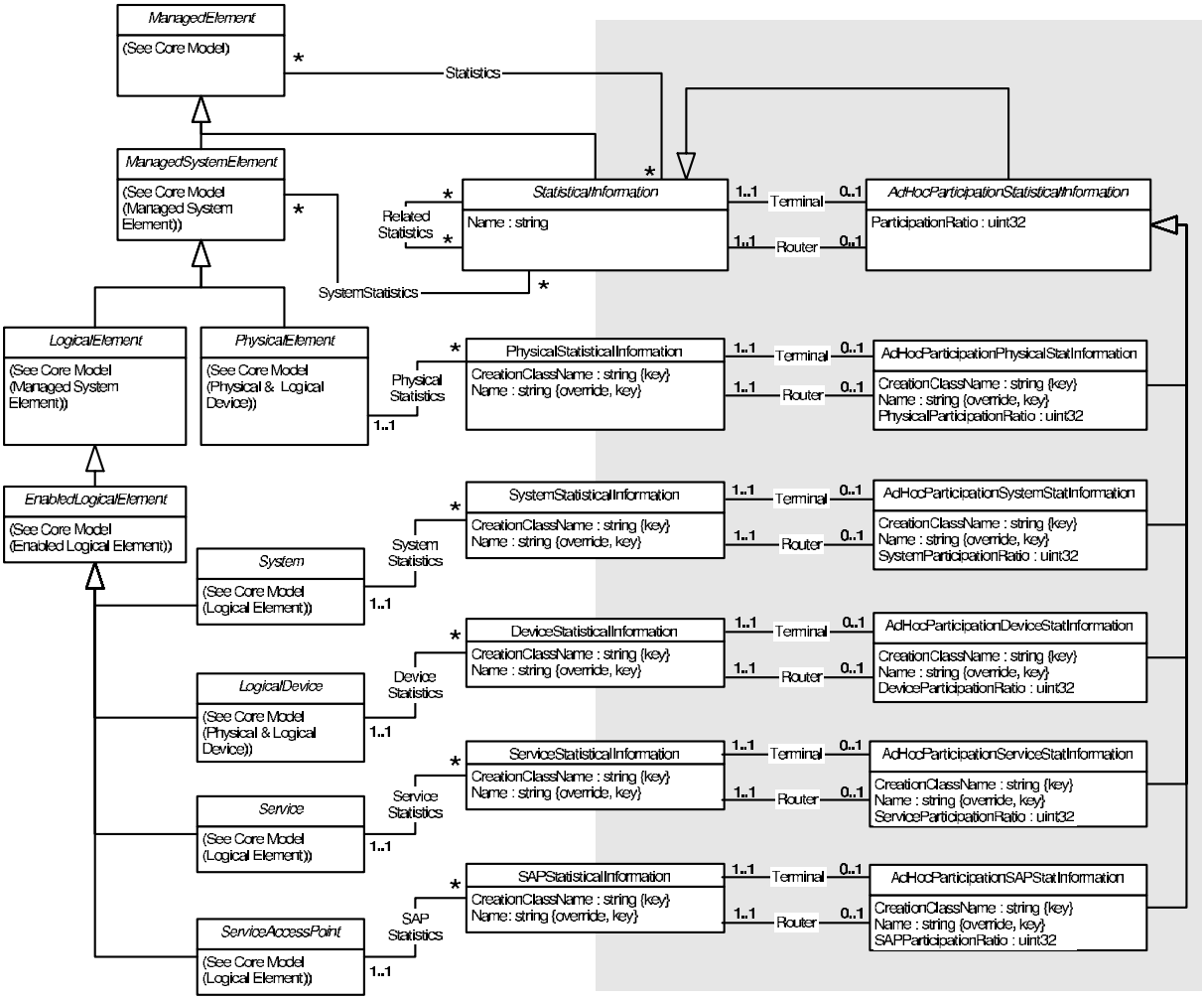


FIG. 4.5 – Diagramme de classes du sous-schéma *Participation*

réseau ad-hoc est modélisée de manière uniforme à travers un ensemble de classes d'extension. Elle peut s'exprimer aussi bien sous la forme de consommation de ressources physiques qu'en terme d'utilisation de services.

En nous appuyant sur le schéma de base et sur le schéma commun, nous avons exprimé les trois principaux sous-schémas de notre modèle d'information qui assure la modélisation des éléments gérés d'un réseau ad-hoc en termes d'organisation, de communication et de participation. Notre démarche vise à offrir une solution suffisamment générique pour être facilement applicable à un très grand nombre de cas. Nous allons désormais spécifier le modèle pour les réseaux ad-hoc implantant le protocole de routage standard OLSR à travers un quatrième sous-schéma.

4.4 Sous-schéma d'extension pour le protocole OLSR

Le protocole OLSR (*Optimized Link State Routing Protocol*) [57] standardisé dans le cadre du groupe de travail MANET de l'IETF [132] est un protocole de routage proactif pour les réseaux ad-hoc. Il s'agit d'une optimisation de l'algorithme à état de liens capable de limiter la consommation en ressources radio lors des diffusions. Comme dans un algorithme à état de liens, chaque nœud réalise deux activités principales : (1) il détermine la liste des nœuds voisins en émettant périodiquement des messages HELLO de beaconing et (2) il échange les informations d'états de liens avec les autres nœuds en diffusant des messages TC (*Topology Control*) de contrôle de topologie, et ce afin de construire et maintenir les tables de routage.

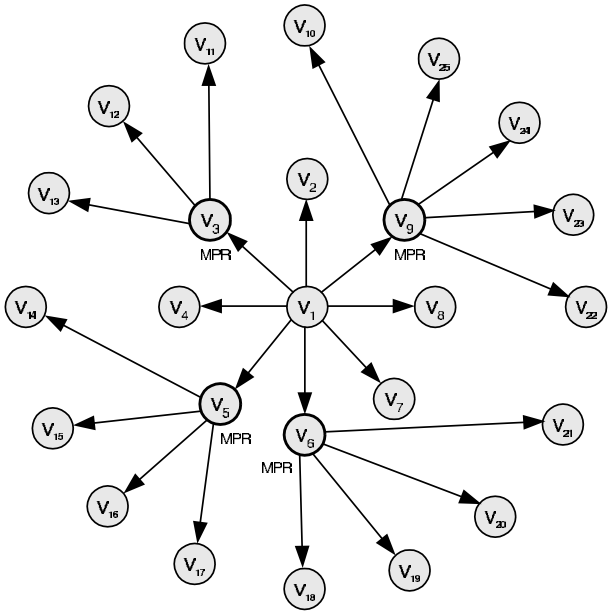


FIG. 4.6 – Sélection de relais multipoints (MPR) avec le protocole OLSR

L'optimisation consiste à limiter le trafic de diffusion en utilisant le concept de relais multipoints [127, 112] : chaque nœud sélectionne un sous-ensemble de ses voisins, appelé MPRs (*Multipoint Relay*), pour retransmettre les paquets lors de la diffusion. Le choix des MPRs est défini par une heuristique qui permet de sélectionner pour un nœud donné le sous-ensemble minimal des voisins à un saut permettant d'atteindre l'ensemble des voisins à deux sauts strictement. Ainsi sur la figure 4.6, le nœud v_1 sélectionne les nœuds voisins à un saut v_3, v_5, v_6 et v_9 comme relais multipoints, pour couvrir l'ensemble des voisins à deux sauts strictement v_{10} à v_{25}.

Nous introduisons un quatrième sous-schéma d'extension pour préciser notre modèle d'information dans le contexte de réseaux ad-hoc implantant le protocole de routage OLSR. La pierre angulaire de ce sous-schéma est représentée par la classe *OLSRAdHocNode* décrite à la figure 4.7. Cette dernière représente un nœud ad-hoc implantant le protocole OLSR : elle hérite de la classe *AdHocNode* et est complétée par un ensemble d'attributs spécifiques au protocole. Elle indique notamment la volonté du nœud à intervenir en tant que relais multipoints lors de la diffusion (attribut *OLSRMPRWillingness*). Elle spécifie également le nombre de relais multipoints couvrant chaque voisin à deux sauts strictement (attribut *OLSRCoverage*), ainsi que la quantité d'informations que le nœud peut inclure dans les messages de diffusion (attri-

but *OLSRTCRedundancy*). Un nœud ad-hoc OLSR peut disposer d'un ensemble d'interfaces OLSR (classe *OLSRIPProtocolEndPoint*), mais l'une d'entre elles est considérée comme l'interface principale (classe d'association *OLSRMainAddress*) permettant d'identifier le nœud. La fréquence d'émission des messages HELLO et TC est fournie par les attributs *OLSRHelloInterval* et *OLSRTCInterval* pour une interface OLSR donnée.

4.4.1 Détection du voisinage

La première activité principale d'un nœud OLSR consiste à évaluer son voisinage à un et deux sauts par l'échange périodique de messages HELLO [94]. Un nœud local doit déterminer les nœuds voisins à un saut avec lesquels il dispose d'un lien direct et symétrique. Il maintient la liste de ses liens locaux (classe d'association *OLSRLocalLinks*), un lien local étant représenté par la classe *OLSRLocalLink* avec l'attribut *OLSRStatus* spécifiant son état (perdu, asymétrique, symétrique). Les liens asymétriques sont également inventoriés, mais ne sont ni utilisés ni propagés tant qu'ils ne sont pas symétriques. Chaque nœud diffuse périodiquement et localement un

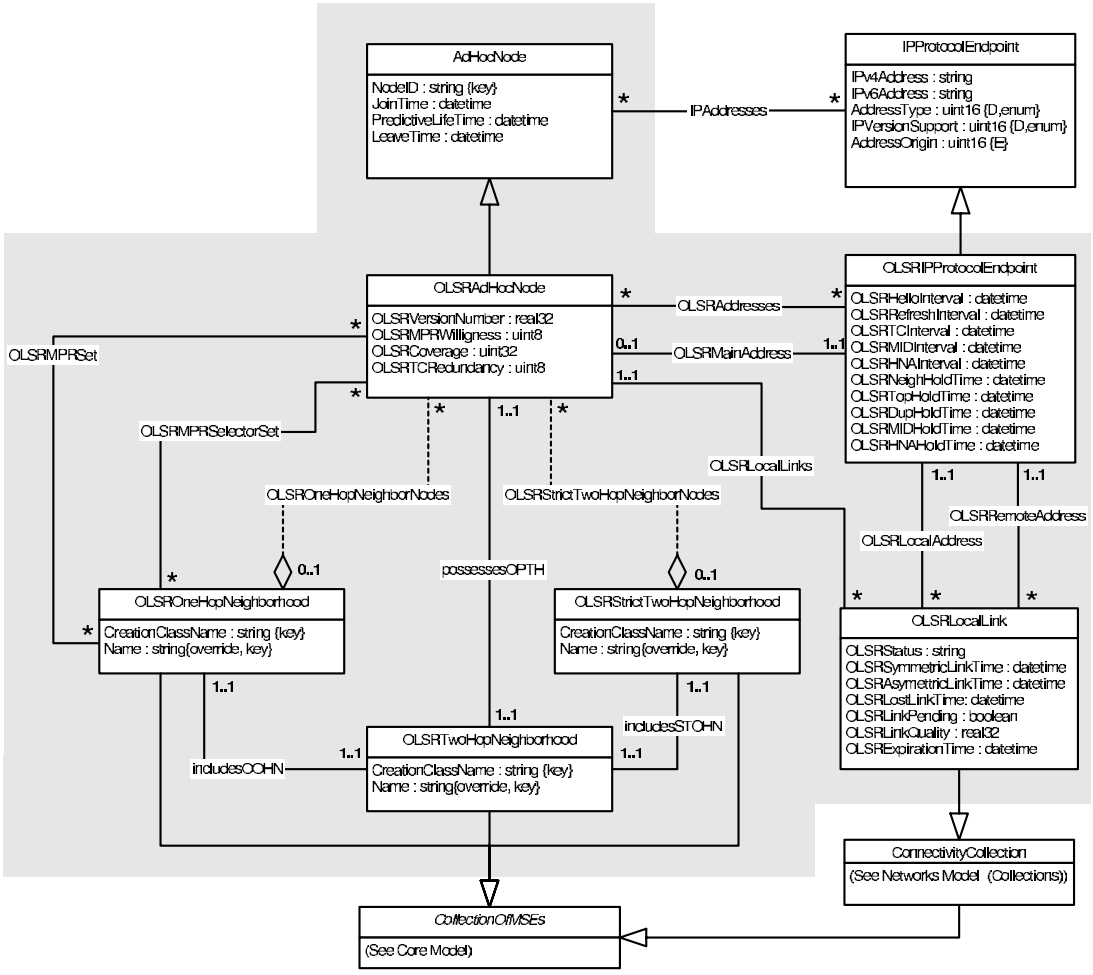


Fig. 4.7 – Diagramme de classes du sous-schéma OLSR portant sur la détection du voisinage et la sélection des relais multipoints

message HELLO contenant des informations sur son voisinage direct et l'état des liens. Ainsi, les nœuds peuvent prendre connaissance de leurs voisinages à un et deux sauts. Par exemple à

la figure 4.6, le nœud v_3 transmet au nœud v_1 des informations sur le voisinage direct, permettant ainsi au nœud v_1 de connaître ses voisins à deux sauts v_{11}, v_{12}, v_{13}. L'ensemble des voisins à un saut et à deux sauts strictement est respectivement défini par les classes d'association *OLSROneHopNeighborhood* et *OLSRStrictTwoHopNeighborhood* sur le diagramme de classes.

4.4.2 Sélection des relais multipoints

En s'appuyant sur ces informations de voisinage, chaque nœud sélectionne la liste de ses relais multipoints correspondant à la classe d'association *OLSRMPRSet*. Les nœuds voisins choisis comme MPR sont annoncés dans les messages HELLO. Ainsi symétriquement, chaque nœud MPR est capable de maintenir la liste des nœuds l'ayant sélectionné comme MPR. Cette liste est représentée par la classe d'association *OLSRMPRSelectorSet* et sera utilisée lors de la diffusion.

4.4.3 Diffusion des informations de topologie

La deuxième activité principale d'un nœud OLSR repose sur l'échange périodique des informations d'états de liens avec les autres nœuds en diffusant des messages TC (*Topology Control*) de contrôle de topologie dans le réseau. Un message TC contient la liste des voisins l'ayant choisi comme MPR et est diffusé dans l'ensemble du réseau. Seuls les nœuds MPR retransmettent un message TC lorsqu'ils le reçoivent et ce afin de réduire la charge de trafic de contrôle lors de la diffusion.

A la réception des messages TC, un nœud OLSR construit la table de topologie. Une entrée de cette table est représentée à la figure 4.8 par la classe *OLSRIPTopologyLink* : elle indique l'adresse du dernier saut (classe d'association *OLSRLastMainAddress*) pour chaque adresse destination (attribut *DestinationAddress*).

4.4.4 Maintenance des tables de routage

Les informations de topologie permettent à chaque nœud OLSR de calculer une table de routage qui permettra l'acheminement des paquets vers n'importe quel nœud destination du réseau. La table de routage indique classiquement pour une adresse destination, l'adresse de prochain saut, l'interface locale considérée et la distance pour atteindre la destination. Une entrée de cette table est symbolisée par la classe *OLSRNextHopIPRoute* de notre modèle d'information. Cette dernière hérite de la classe générique *NextHopIPRoute* définie dans le schéma commun.

4.4.5 Prise en charge des interfaces multiples

Un nœud OLSR peut disposer d'interfaces multiples sur lesquelles sont exécutées le protocole OLSR, mais il sélectionne toujours une interface principale pour son identification. La prise en charge des interfaces multiples est assurée par la diffusion de messages MID (*Multiple Interface Declaration*) permettant de déclarer des interfaces supplémentaires.

En particulier, à la réception de messages MID, chaque nœud OLSR construit une table d'associations. La table indique pour chaque interface multiple OLSR quelle est l'interface OLSR principale du nœud. Une entrée de la table est représentée par la classe *OLSRMultipleInterfaceAssociation* sur le diagramme de la figure 4.8.

4.4.6 Prise en charge des interfaces non OLSR

Un nœud OLSR peut également posséder des interfaces non OLSR : il peut typiquement être connecté à un réseau fixe par le biais d'une interface ethernet. Pour prendre en charge les interfaces non OLSR, le protocole de routage gère une seconde table d'associations. Le fonctionnement de cette table est similaire à la précédente, à la différence qu'elle met en correspondance l'interface OLSR principale du nœud avec l'adresse d'un réseau ou d'un hôte externe. Une en-

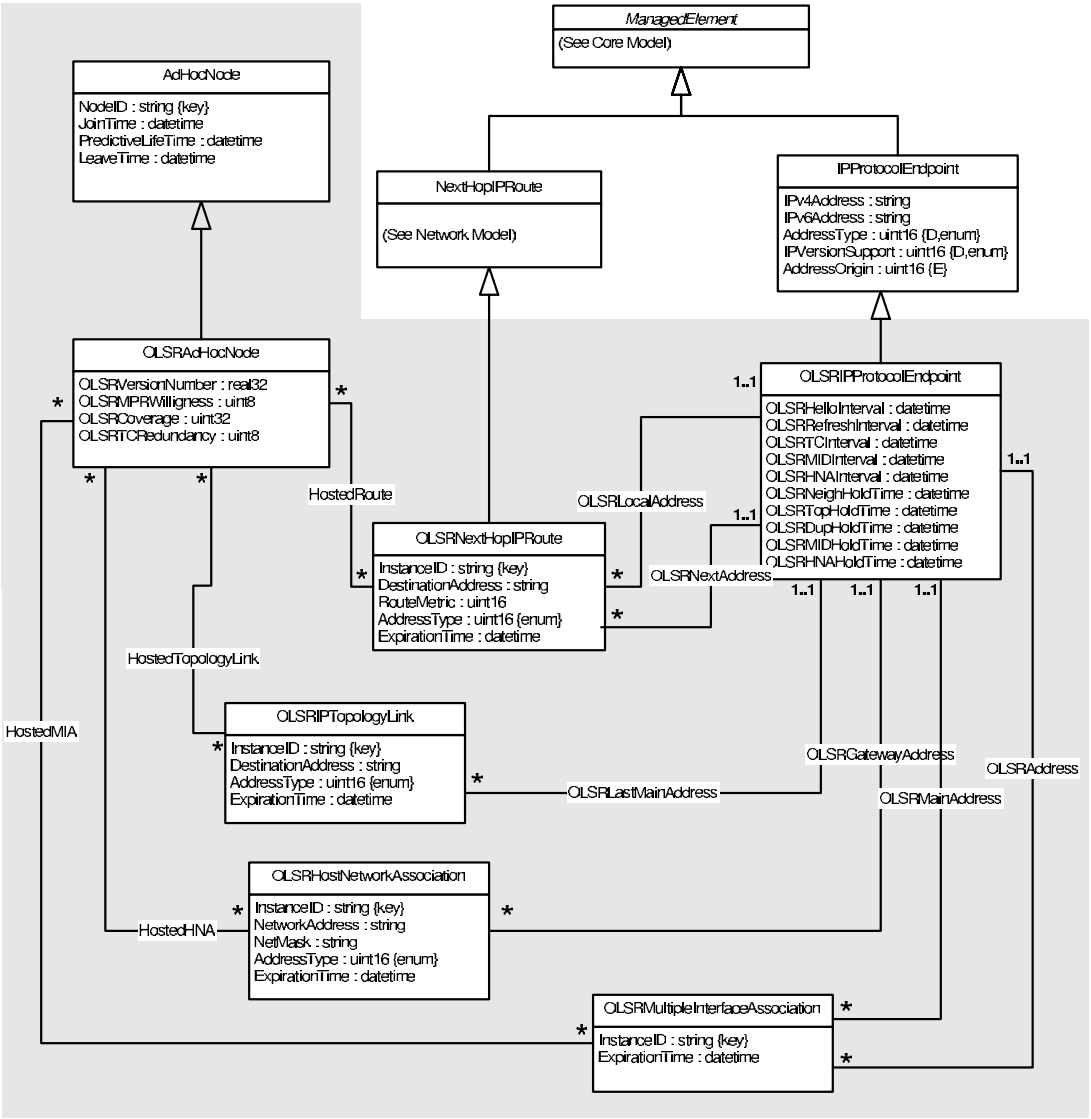


FIG. 4.8 – Diagramme de classes du sous-schéma OLSR portant sur la diffusion de topologie, la maintenance des tables de routage et la prise en charge des interfaces multiples et non OLSR

trée de la table est représentée par la classe *OLSRHostNetworkAssociation* sur le diagramme de classes. La classe d'association *OLSRGatewayAddress* indique l'interface principale du nœud OLSR disposant d'une interface externe (non OLSR).

4.5 Synthèse

La modélisation de l'information est une étape essentielle pour permettre l'intégration des réseaux et services ad-hoc dans une démarche de gestion. L'objectif consiste à identifier ce qui peut être observé et géré dans un tel contexte et à spécifier ces informations dans un formalisme commun. Le modèle commun de l'information (CIM) est devenu un standard incontournable pour modéliser les ressources d'un environnement géré.

Nous avons présenté dans ce chapitre une extension de ce modèle pour prendre en charge les réseaux et services ad-hoc. Notre extension traduit les caractéristiques essentielles des réseaux ad-hoc. Nous avons notamment détaillé (1) l'organisation du réseau ad-hoc, typiquement sous la forme de clusters, (2) les échanges au sein du réseau à différentes échelles et (3) la participation des entités au fonctionnement du réseau, tant en terme de consommation de ressources physiques qu'en terme d'utilisation de services.

Par ailleurs, la standardisation récente des protocoles de routage ad-hoc nous permet de compléter notre modèle d'information. Les protocoles de routage tels que BGP et OSPF sont déjà largement intégrés au modèle CIM. En suivant une démarche similaire, nous avons précisé le modèle pour assurer la prise en charge d'un protocole de routage ad-hoc standard, en l'occurrence le protocole de routage proactif OLSR. Nous avons également dérivé de cette spécification une base d'informations de gestion nommée OLSR-MIB que nous présentons en annexe.

Après s'être attaché à l'objet de la gestion en spécifiant le modèle d'information, il est important d'en définir les moyens, en particulier définir comment organiser le plan de gestion dans un tel environnement dynamique.

Les travaux présentés dans ce chapitre ont fait l'objet de plusieurs publications [13, 18].

Chapitre 5

Organisation probabiliste du plan de gestion

Sommaire

5.1 Introduction

Une approche de gestion au sens pur du terme, où l'ensemble des nœuds serait géré à tout instant, est trop stricte pour les réseaux ad-hoc. Au lieu de considérer la gestion du réseau dans son intégralité, nous proposons dans ce chapitre une nouvelle organisation du plan de gestion fondée sur une méthode probabiliste [109, 8]. Cette dernière permet de restreindre volontairement l'activité de gestion en la limitant à des sous-ensembles de nœuds, afin de fournir une solution légère et performante. Les nœuds sont sélectionnés en fonction de leurs comportements et de leurs dépendances, l'objectif étant de favoriser ceux qui ont à la fois une forte présence dans le réseau et une grande connectivité. En s'appuyant sur cette approche sélective, nous dérivons des garanties sur le pourcentage de nœuds qui seront actifs dans le plan de gestion.

Ce travail de recherche représente un axe orthogonal à celui de la modélisation de l'information. Nous avons traité, dans le chapitre précédent, l'identification de ce qui peut être géré dans un réseau ad-hoc, en laissant pour compte la manière avec laquelle nous allions assurer cette gestion. Nous complétons notre travail en définissant comment organiser le plan de gestion de manière efficace. Le fondement premier de notre démarche est de relâcher les contraintes sur le plan de gestion en ne considérant que certains nœuds à gérer. Cette démarche est tout à fait appropriée au cadre des réseaux ad-hoc : de manière générale, seuls les nœuds qui ont une présence effective dans le réseau ad-hoc présentent un réel intérêt pour la supervision.

Notre contribution correspond à l'application d'une démarche probabiliste pour l'organisation du plan de gestion. Elle inclut la formalisation de cette démarche à travers une méthode algorithmique de gestion définie au niveau applicatif, son déploiement au sein d'une architecture de gestion et son évaluation à travers un ensemble d'expérimentations. La méthode algorithmique permettra à la fois d'extraire les nœuds ad-hoc à gérer en fonction de leurs propriétés spatiales et temporelles, et de déterminer les nœuds gestionnaires en appliquant différents mécanismes électifs reposant sur la notion de centralité.

Dans ce chapitre, nous introduirons tout d'abord le concept de la gestion probabiliste pour les réseaux ad-hoc. Puis, nous détaillerons la méthode algorithmique de gestion en décrivant la sélection des nœuds prenant part au plan de gestion et l'élection de nœuds gestionnaires. Nous présenterons différents mécanismes électifs reposant sur la centralité de degré et la centralité par vecteur propre. Nous montrerons ensuite comment intégrer l'approche probabiliste dans l'architecture de gestion ANMP, avant de terminer par une analyse des résultats expérimentaux obtenus par la simulation.

5.2 Gestion probabiliste des réseaux ad-hoc

Notre approche probabiliste repose sur la simple constatation que la gestion d'un réseau ad-hoc peut difficilement prendre en charge la totalité des nœuds et peut difficilement s'efxécuter de façon continue (comme cela peut être la cas dans les réseaux fixes traditionnels). En particulier, si le comportement d'un nœud ad-hoc peut être décrit de manière stochastique (un nœud étant présent dans le réseau avec une certaine probabilité), pourquoi devrions-nous conserver un cadre déterministe pour la gestion de réseaux ?

Modélisation stochastique

Les travaux de Burgess [40] sont les premiers à avoir proposé une modélisation des réseaux ad-hoc dans le cadre de la gestion. Les réseaux ad-hoc y sont définis à l'aide d'une matrice d'adjacence contenant pour chaque entrée (i, j) la probabilité que les nœuds v_i et v_j soient voisins directs. Notre travail est partiellement motivé par cette approche : si nous supposons qu'une telle matrice est connue ou partiellement connue (à travers les informations de topologie d'un protocole de routage), pourquoi ne pas restreindre le plan de gestion aux nœuds qui semblent avoir une présence effective et qui intercommuniquent au sein du réseau ? Pourquoi, également, ne pas définir une méthode probabiliste dans ce cadre ? Cette formalisation du plan de gestion nous permettra de dériver des garanties sur le pourcentage de nœuds gérés. Dans un réseau fixe, les relations directes entre voisins évoluent sur des échelles de temps relativement longues et la gestion tente d'administrer l'intégralité des nœuds du réseau. Dans un réseau ad-hoc, nous n'assurerons pas que tous les nœuds seront gérés mais offrirons des seuils statistiques quant au pourcentage de nœuds ad-hoc pris en charge par la gestion.

Restriction volontaire du plan de gestion

Une méthode probabiliste a déjà été mise en œuvre au niveau du modèle d'information dans [68, 67, 66]. L'objectif était d'améliorer l'acquisition des informations de monitorage dans un réseau ad-hoc et notamment de déterminer statistiquement les valeurs qui n'ont pas pu être mesurées par les équipements mobiles. Les informations brutes n'étaient pas directement accessibles mais devaient être traitées auparavant par un modèle probabiliste permettant la corrélation des données.

Pour accéder au modèle d'information, une requête sur les données devait être formulée, puis être transmise à un module d'acquisition. Ce module implantait le modèle probabiliste qui corrélait les informations brutes en admettant une distribution normale des variables observées et fournissait après ce traitement des données fiables. Si les données brutes sont disparates ou biaisées, elles peuvent conduire à une mauvaise connaissance de l'environnement. Les mécanismes statistiques permettaient d'offrir une approximation de l'information avec des probabilités de confiance. L'efficacité de l'approche dépendait directement des modèles probabilistes considérés pour une variable donnée.

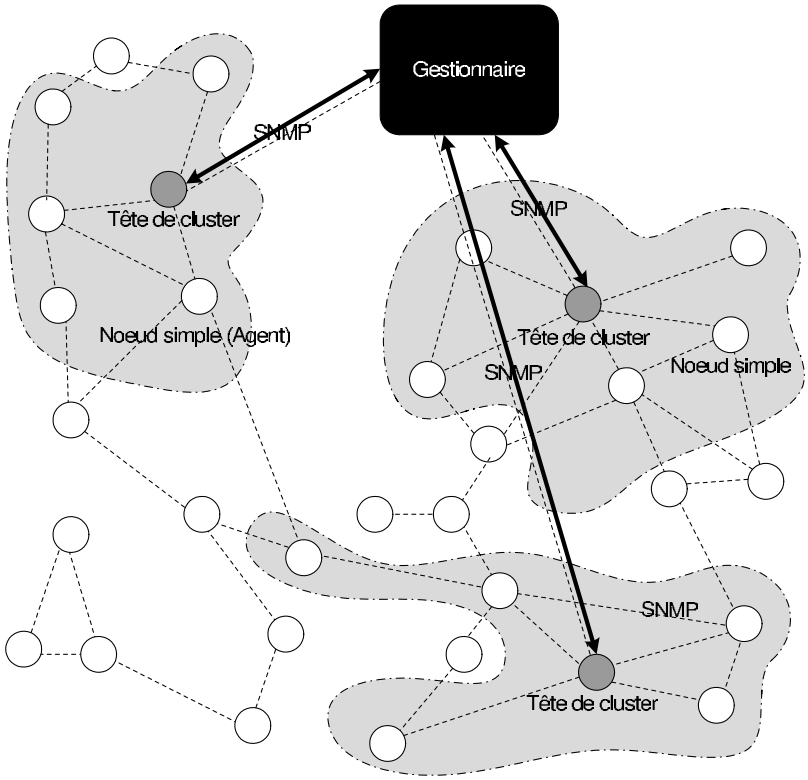


Fig. 5.1 – Organisation probabiliste du plan de gestion

Nous proposons de mettre en œuvre une méthode probabiliste, mais cette fois-ci au niveau du modèle organisationnel en analysant les relations entre pairs. Nous souhaitons limiter le plan de gestion à un ou plusieurs sous-ensembles de nœuds, de sorte que nous sélectionnions les nœuds les plus *intéressants* en terme de gestion. Ainsi sur la figure 5.1, nous constatons que le plan de gestion n'intègre plus tous les nœuds mais se limite à ceux présents dans les zones grisées. Nous entendons par *intéressants* des nœuds qui disposent d'une présence effective et d'un positionnement stratégique dans la topologie du réseau.

Nous introduisons pour ce faire la notion de composante spatio-temporelle et la définissons comme un sous-ensemble de nœuds du réseau où chaque nœud dispose d'une forte probabilité d'adjacence. Le terme spatio-temporel traduit à la fois la dimension spatiale de la composante (les nœuds doivent être adjacents) et sa dimension temporelle (les nœuds doivent être adjacents sur une période de temps suffisante). Nous restreignons le plan de gestion aux composantes spatio-temporelles les plus larges et considérons que seuls les nœuds appartenant à ces composantes seront gérés. Cette approche sélective permet de limiter la charge de gestion pour les réseaux ad-hoc. Nous allons détailler dans la section suivante la méthode algorithmique correspondante qui permet cette nouvelle organisation du plan de gestion.

5.3 Méthode algorithmique de gestion

La méthode algorithmique implantant notre approche de gestion probabiliste est fondée sur une mesure de connectivité spatio-temporelle. Un nœud du réseau évalue la connectivité spatio-temporelle qu'il entretient avec ses voisins et communique l'information aux autres nœuds du réseau afin d'établir conceptuellement la matrice de connectivité spatio-temporelle du réseau ad-hoc. Nous montrerons par la suite que cette matrice peut se rapprocher, après un traitement, des informations de topologie dont dispose un nœud ad-hoc implantant un protocole de routage proactif tel que le protocole OLSR. A partir de cette matrice, le nœud est capable de déterminer localement les composantes spatio-temporelles du réseau et de sélectionner un ou plusieurs nœuds gestionnaires. Cette méthode est définie au niveau applicatif dans le plan de gestion. Nous supposons une coopération minimale entre les nœuds ainsi qu'un contrôle partiel de ceux-ci. Si nécessaire, la coopération peut être stimulée à l'aide d'une approche incitative telle que présentée dans [43].

5.3.1 Mesure de connectivité spatio-temporelle

Un réseau ad-hoc peut être défini comme un ensemble de n nœuds mobiles $V = \{v_1, v_2, ..., v_n\}$ se déplaçant sur une surface donnée durant une période de temps T. La période de temps T est décomposée en k intervalles de mesure $[t_l, t_{l+1}]$ avec $t_l = l \times \frac{T}{k}$ pour un entier $l \in [0, k]$. Le choix d'une période de temps optimal T ainsi que les valeurs des intervalles $[t_l, t_{l+1}]$ ne sont pas discutés dans ce cadre mais une excellente analyse des échelles de temps pour le flux d'information de gestion peut être trouvée dans [71]. Chaque nœud évalue la connectivité spatio-temporelle qu'il entretient avec son voisinage. Cette mesure notée $m_{STC}(i, j)$ correspond pour un noeud v_i au pourcentage de temps durant lequel il a été voisin d'un nœud v_j du réseau. Sur un intervalle de temps $[t_l, t_{l+1}]$, les mesures sont stockées localement dans une liste de valeurs $N^l(v_i)$ composée de tuples $(v_i, v_j, m_{STC}(i, j))$ et sont ensuite échangées et fusionnées entre les nœuds du réseau. Le suffixe l représente le facteur temps et signifie que la mesure a été réalisée dans l'intervalle de temps $[t_l, t_{l+1}]$.

L'échange des mesures locales conduit conceptuellement à la matrice de connectivité spatio-temporelle notée M^l_{STC} (qui fournit une vue à l'échelle du réseau) obtenue en concaténant la liste des mesures réalisées localement par les nœuds. Chaque ligne/colonne représente un nœud du réseau. La i-ème ligne de la matrice M^l_{STC} représente la liste des mesures $N^l(v_i)$ du nœud v_i. Si un nœud v_i a été voisin du nœud v_j sur l'intervalle $[t_l, t_{l+1}]$, alors une entrée $M^l_{STC}[i, j]$ existe dans la matrice et contient la mesure $m_{STC}(i, j)$ du pourcentage de temps durant lequel la paire de nœuds (v_i, v_j) a été directement connectée sur cet intervalle. Comme l'objectif est d'identifier des nœuds présentant une forte probabilité d'adjacence et également pour limiter les

quantités de données gérées, seules les valeurs de connectivité spatio-temporelle dépassant une valeur seuil donnée seront considérées lors de l'extraction des composantes.

5.3.2 Extraction de composantes spatio-temporelles

La matrice M^l_{STC} peut être représentée sous la forme d'un graphe composé de n nœuds tel que montré à la figure 5.2. Un lien existe entre deux nœuds du graphe si les deux nœuds ont été voisins dans l'intervalle de temps $[t_l, t_{l+1}]$. Ce lien est pondéré par la mesure de connectivité spatio-temporelle $m_{STC}(i, j)$. Chaque nœud v_i peut déterminer la composante spatio-temporelle (notée CC_{v_i}) à laquelle il fait partie en parcourant le graphe. Ce parcours est limité aux liens correspondant à une connectivité spatio-temporelle minimale (supérieure à une valeur seuil λ prédéfinie). Une composante spatio-temporelle correspond ainsi à un sous-ensemble de nœuds connectés de l'ensemble V. L'identification des composantes spatio-temporelles est décrite par l'algorithme 1 et correspond à l'extraction des composantes connexes du graphe [61, 93] limitée aux liens les plus significatifs dans le temps.

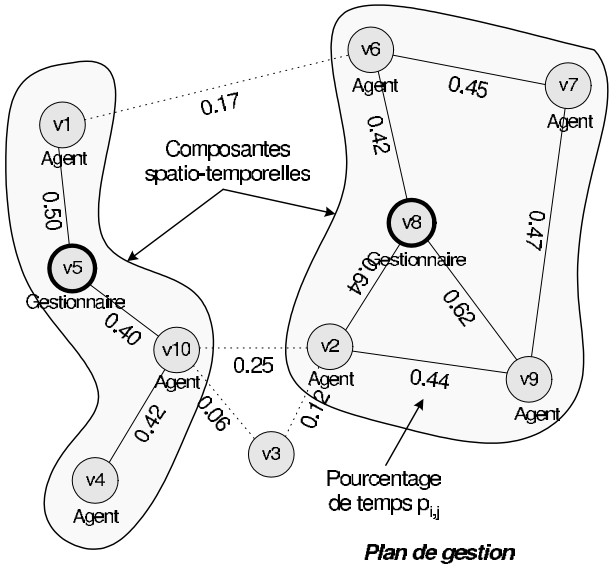


FIG. 5.2 – Extraction des composantes spatio-temporelles dans le plan de gestion

Le sous-ensemble CC_{v_i} est initialisé avec un unique élément : le nœud ad-hoc v_i lui-même. L'algorithme permet ensuite d'extraire les composantes connexes en ajoutant itérativement à CC_{v_i} les nœuds voisins de chaque nœud faisant déjà partie du sous-ensemble CC_{v_i}. Un nœud ad-hoc v_i peut extraire la composante spatio-temporelle CC_{v_i} à laquelle il est associé, mais également il peut extraire les autres composantes du réseau CC_{v_j} $(j \neq i)$ en appliquant la même démarche, mais dans ce cas en initialisant le nouveau sous-ensemble avec un nœud qui ne fait pas encore partie de sa composante. Chaque élément d'une composante spatio-temporelle a été connecté à un autre élément de la composante durant au moins un pourcentage de temps minimal.

Si nous considérons l'exemple présenté à la figure 5.2, le nœud v_2 détermine sa composante spatio-temporelle de la manière suivante. L'ensemble CC_{v_2} est initialisé avec le nœud v_2. À la première itération de l'algorithme, les nœuds voisins v_8 et v_9 sont ajoutés à CC_{v_2}. À la deuxième itération, les nœuds v_6 et v_7, voisins respectifs de v_8 et v_9 sont également ajoutés à la composante

Algorithme 1 : Extraction des composantes spatio-temporelles

Données : Matrice de connectivité spatio-temporelle M^l_{STC}
Résultat : Composante spatio-temporelle CC_{v_i} du nœud v_i
initialisation
a) initialiser l'ensemble CC_{v_i} avec le nœud $\{v_i\}$ comme élément unique ;
$CC_{v_i} = \{v_i\}$
répétition
b) ajouter à l'ensemble CC_{v_i} tous les nœuds connectés à un nœud élément de CC_{v_i} ;
$\forall x \in CC_{v_i},\ y\ voisin\ de\ x\ \wedge\ M^l_{STC}(x,y) > \lambda \Rightarrow CC_{v_i} = CC_{v_i} \cup \{y\}$
c) supprimer les doublons éventuels de CC_{v_i} ;
jusqu'à
d) l'ensemble CC_{v_i} reste inchangé entre deux itérations.

spatio-temporelle. La troisième itération n'entraîne aucune modification de l'ensemble et clot ainsi l'exécution de l'algorithme.

5.3.3 Election de gestionnaires par analyse de la centralité

Une composante spatio-temporelle telle que CC_{v_i} représente un sous-ensemble de nœuds caractérisé par une forte valeur de connectivité spatio-temporelle. Cela constitue en fait un sous-domaine de gestion qui pourra être à la charge d'un ou plusieurs nœuds gestionnaires en fonction de la taille de la composante. Un mécanisme d'élection est requis pour déterminer les nœuds gestionnaires pour chacune des composantes spatio-temporelles.

En considérant que chaque nœud ad-hoc dispose d'un unique identifiant (par exemple, son adresse MAC), la solution la plus simple est de considérer une élection arbitraire fondée sur cet identifiant. Il est typiquement possible d'élire comme gestionnaire le nœud disposant du plus petit identifiant. La figure 5.3 décrit une composante spatio-temporelle composée de neuf nœuds identifiés à l'aide d'un indice unique. Si nous considérons cet indice comme critère d'élection, le nœud v_1 dispose du plus petit identifiant et sera alors sélectionné comme le gestionnaire de la composante, même si ce nœud est relativement isolé par rapport aux autres nœuds : il dispose d'un unique lien vers le nœud v_3. Le mécanisme d'élection n'est pas efficace car il ne prend en compte ni les propriétés structurelles de la composante spatio-temporelle, ni l'importance relative des nœuds.

Notion de centralité

Nous proposons deux mécanismes plus rafinés fondés sur la notion de centralité de nœuds dans un graphe [85]. La centralité quantifie l'importance d'un nœud dans une structure. Elle nous permet d'identifier les nœuds qui disposent d'une position centrale et qui ont donc *a priori* une influence plus importante sur les autres membres [86]. Les nœuds présentant une forte centralité seront sélectionnés comme nœuds gestionnaires. Les mesures de centralité seront effectuées à partir des données formées par l'ensemble des valeurs de connectivité spatio-temporelle limité aux nœuds présents dans la composante. Cet ensemble de données correspond à une sous-matrice de M^l_{STC} que nous noterons S^l_{STC} où chaque ligne/colonne représente l'un des m nœuds de la composante. Si nous considérons toujours le scénario présenté à la figure 5.3, alors la composante est formée de neuf nœuds ad-hoc et est associée à la sous-matrice 9×9 S^l_{STC} suivante.

$$S^l_{STC} = \begin{bmatrix} |0 & 0 & 0.42 & 0 & 0 & 0 & 0 & 0 & 0 \\ |0 & 0 & 0.54 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.42 & 0.54 & 0 & 0.63 & 0 & 0.57 & 0 & 0 & 0 \\ 0 & 0 & 0.63 & 0 & 0.64 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.64 & 0 & 0.62 & 0 & 0 & 0 \\ 0 & 0 & 0.57 & 0 & 0.62 & 0 & 0.71 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.71 & 0 & 0.62 & 0.48 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.62 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.48 & 0 & 0 \end{bmatrix}$$

La sous-matrice S^l_{STC} représente un graphe non orienté et correspond alors à une matrice symétrique. Différentes variantes de la mesure de centralité ont été définies dans les réseaux sociaux [34]. Nous allons détailler deux de ces variantes : la centralité de degré et la centralité par vecteur propre, qui nous serviront de support pour élire les nœuds gestionnaires des composantes spatio-temporelles.

Mécanisme électif fondé sur la centralité de degré

La centralité de degré est une forme de centralité les plus simples à mettre en œuvre puisqu'elle correspond au degré du nœud [69]. La centralité de degré est calculée comme le nombre de liens attachés à un nœud dans un graphe non orienté. Un nœud est d'autant plus considéré comme central qu'il est directement connecté à un grand nombre d'autres nœuds. Chaque lien étant pondéré par un pourcentage de temps, la centralité de degré est calculée comme suit à l'aide l'équation 5.1. Nous calculons pour chaque nœud v_i sa centralité de degré notée $c_{deg}(v_i)$ comme la somme des valeurs en pourcentage de temps de chaque lien connecté au nœud considéré.

$$c_{deg}(v_i) = \sum_{j=1}^{j=m} S^l_{STC}(i,j) \ pour\ un\ noeud\ v_i\ donné \tag{5.1}$$

Sous forme matricielle, la mesure de centralité d'un nœud v_i est calculée comme la somme des éléments de la i-ème ligne de la sous-matrice S^l_{STC}, ce qui revient à multiplier la sous-matrice avec le vecteur unité $\vec{1}$ comme dépeint par l'équation 5.2.

$$\overrightarrow{c_{deg}} = S^l_{STC} \times \vec{1}\ (forme\ matricielle) \tag{5.2}$$

Nous appliquons le mécanisme d'élection dans le cadre du scénario présenté à la figure 5.3 et nous aboutissons au vecteur $\overrightarrow{c_{deg}}$ contenant pour chaque nœud v_i sa centralité de degré à la i-ème entrée.

$$\overrightarrow{c_{deg}} = [\ 0.42\ 0.54\ 2.16\ 1.27\ 1.26\ 1.90\ 1.81\ 1.62\ 0.48\ ]^T \tag{5.3}$$

Le classement des valeurs est synthétisé dans le premier tableau de la figure 5.3 où nous pouvons constater que le nœud v_3 présente le plus fort degré avec une valeur de 2.16. Ce nœud pourra alors être sélectionné comme le gestionnaire de la composante spatio-temporelle.

Mécanisme électif fondé sur la centralité par vecteur propre

La centralité de degré mesure localement le nombre de nœuds voisins à un saut, indépendamment de la centralité des nœuds auxquels le nœud est directement lié. Or, intuitivement, un nœud lié à des nœuds isolés n'est pas aussi central que celui qui est lié à des nœuds eux-mêmes

centraux. Une mesure alternative est la centralité par vecteur propre proposée par Bonacich dans [32]. L'objectif est de définir une mesure de centralité de manière récursive : un nœud est d'autant plus central qu'il est connecté à des nœuds qui sont eux-mêmes centraux. Ainsi, la centralité $c_{eig}(v_i)$ dépend de la centralité des nœuds voisins comme décrit dans l'équation 5.4.

$$c_{eig}(v_i) = \frac{1}{\lambda}(m_{STC}(i,1).c_{eig}(v_1) + ... + m_{STC}(i,j).c_{eig}(v_j) + ... + m_{STC}(i,m).c_{eig}(v_m)) \quad (5.4)$$

Pour un tel graphe non orienté, Bonacich démontre que le problème peut être réduit au problème classique du vecteur propre principal [63] défini par l'équation 5.5.

$$\lambda \times \vec{x} = S^l_{STC} \times \vec{x} \ (forme\ matricielle) \quad (5.5)$$

Les solutions de l'équation sont définies par des couples (vecteur propre, valeur propre). Le vecteur propre principal $\vec{v}_{pr}$ est le vecteur solution qui est associé à la plus grande valeur propre. La centralité par vecteur propre c_{eig} du nœud v_i correspond au i-ème élément du vecteur propre principal de la matrice S^l_{STC} comme précisé à l'équation 5.6.

$$c_{eig}(v_i) = \vec{v}_{pr}(i) \ pour\ un\ noeud\ donné\ v_i \quad (5.6)$$

Nous avons calculé la centralité par vecteur propre avec le scénario décrit dans la figure 5.3 en calculant le vecteur propre principal à l'aide de l'algorithme 2 (où $\vec{v}^k_{pr}$ représente l'approximation du vecteur propre principal $\vec{v}_{pr}$ à la k-ième itération). Cet algorithme aboutit au vecteur propre

Algorithme 2 : Election de gestionnaires par calcul de la centralité par vecteur propre

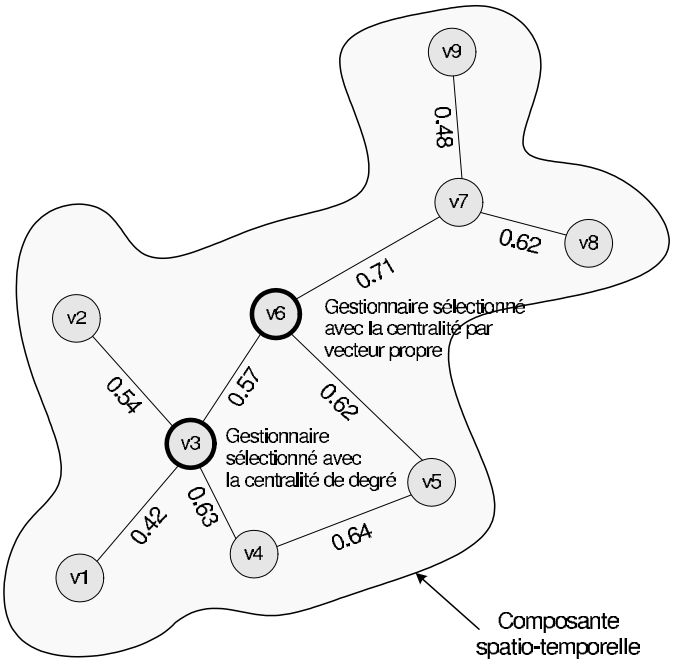
Données : Sous-matrice de connectivité spatio-temporelle S^l_{STC}
Résultat : Vecteur propre principal $\vec{v}_{pr}$
initialisation
a) initialiser le vecteur solution avec des valeurs identiques pour chaque entrée ;
$\vec{v}^0_{pr} = \frac{1}{m} \times \vec{1}$
répétition
b) calculer le nouveau vecteur $\vec{v}_{pr}$ en appliquant la matrice de connectivité S^l_{STC} ;
$\vec{v}^{k+1}_{pr} = S^l_{STC} \times \vec{v}^k_{pr}$
c) normaliser le vecteur $\vec{v}_{pr}$ par la somme de ses coefficients ;
d) estimer la convergence en comparant $\vec{v}^k_{pr}$ et $\vec{v}^{k+1}_{pr}$;
$\delta = \| \vec{v}^{k+1}_{pr} - \vec{v}^k_{pr} \|$
jusqu'à
e) la convergence est effective i.e. $\delta < \epsilon$

décrit à l'équation 5.7. Le i-ème élément de ce vecteur nous fournit la centralité du nœud v_i.

$$\vec{v}_{pr} = [\ 0.14\ 0.17\ 0.47\ 0.37\ 0.39\ 0.52\ 0.36\ 0.15\ 0.12\ ]^T \quad (5.7)$$

Les éléments du vecteur propre principal sont ensuite classés par ordre d'importance afin de construire le second tableau de la figure 5.3. Nous constatons dans ce tableau que le nœud v_6 présente l'entrée la plus élevée avec une centralité par vecteur propre de 0.52 et pourra donc être sélectionné comme nœud gestionnaire.

Sur la figure 5.3, les deux tableaux représentent respectivement les valeurs pour chacun des mécanismes électifs appliqués à la composante spatio-temporelle. Pour chaque tableau, les nœuds sont classés par ordre d'importance, de sorte que les premières entrées correspondent toujours



Noeud	Centralité de degré
v3	2.16
v6	1.90
v7	1.81
v4	1.27
v5	1.26
v8	1.62
v2	0.54
v9	0.48
v1	0.42

Noeud	Centralité par vecteur propre
v6	0.52
v3	0.47
v5	0.39
v4	0.37
v7	0.36
v2	0.17
v8	0.15
v1	0.14
v9	0.12

FIG. 5.3 – Elections de nœuds gestionnaires dans une composante spatio-temporelle [la centralité par degré et la centralité par vecteur propre aboutissent à des résultats différents présentés respectivement dans le premier et second tableau. Le premier mécanisme conduit à l'élection du nœud v_3 tandis que le second sélectionne le nœud v_6.]

aux nœuds les plus importants qui seront sélectionnés comme gestionnaires. L'élection fondée sur la centralité de degré élit le nœud v_3 comme gestionnaire, tandis que l'élection fondée sur la centralité par vecteur propre sélectionne le nœud v_6. Nous constatons que les deux approches de centralité n'aboutissent pas au même résultat : la centralité par degré est une mesure purement locale tandis que la centralité par vecteur propre prend en compte récursivement l'importance relative des nœuds voisins. Une version généralisée de la centralité par vecteur propre est définie dans [33] pour traiter le cas des graphes orientés.

5.4 Intégration à l'architecture de gestion ANMP

Nous proposons de mettre en œuvre notre approche probabiliste au sein d'une architecture de gestion en implantant la méthode algorithmique précédemment décrite. En particulier, nous allons définir une extension de l'architecture ANMP (*Ad-Hoc Network Management Protocol*) [53] décrite à la figure 5.4. Si ANMP définit un protocole de gestion pour les réseaux ad-hoc, le concept clé repose sur l'organisation du plan de gestion sous la forme de clusters et l'introduction de mécanismes de délégation [197] afin de réduire la charge induite sur le réseau et améliorer la tolérance aux fautes. Le plan de gestion est hiérarchisé en trois niveaux correspondant à un gestionnaire central au niveau supérieur, des gestionnaires locaux correspondant aux têtes de clusters au niveau intermédiaire et des agents correspondant aux autres nœuds des clusters au niveau inférieur.

5.4.1 Ajout d'un module de clusterisation

Former des clusters est la manière la plus naturelle de décomposer le réseau ad-hoc afin d'en simplifier les tâches de gestion. Une telle décomposition peut s'opérer de manières différentes en fonction de l'algorithme considéré. ANMP introduit deux algorithmes de clusterisation définis au niveau applicatif permettant d'organiser le plan de gestion :

- algorithme de clusterisation fondé sur les graphes : il construit des clusters de voisins à un saut en s'appuyant sur un mécanisme électif arbitraire. Chaque nœud du réseau ad-hoc maintient une liste de son voisinage direct et détermine le gestionnaire local en fonction des identifiants. Le nœud gestionnaire correspond au nœud qui dispose à la fois du plus petit identifiant et n'a pas rejoint un autre cluster.
- algorithme de clusterisation géographique : il définit des clusters de voisins jusqu'à trois sauts en utilisant un système de positionnement [98, 150, 191]. À partir des coordonnées des nœuds, l'algorithme considère la densité de nœuds dans le réseau pour organiser les clusters en conséquence. Une zone de forte densité sera divisée en de multiples clusters et sera donc gérée par plusieurs gestionnaires locaux. Inversement, une zone de faible densité pourra se limiter à un unique cluster et ainsi être sous la responsabilité d'un unique gestionnaire local.

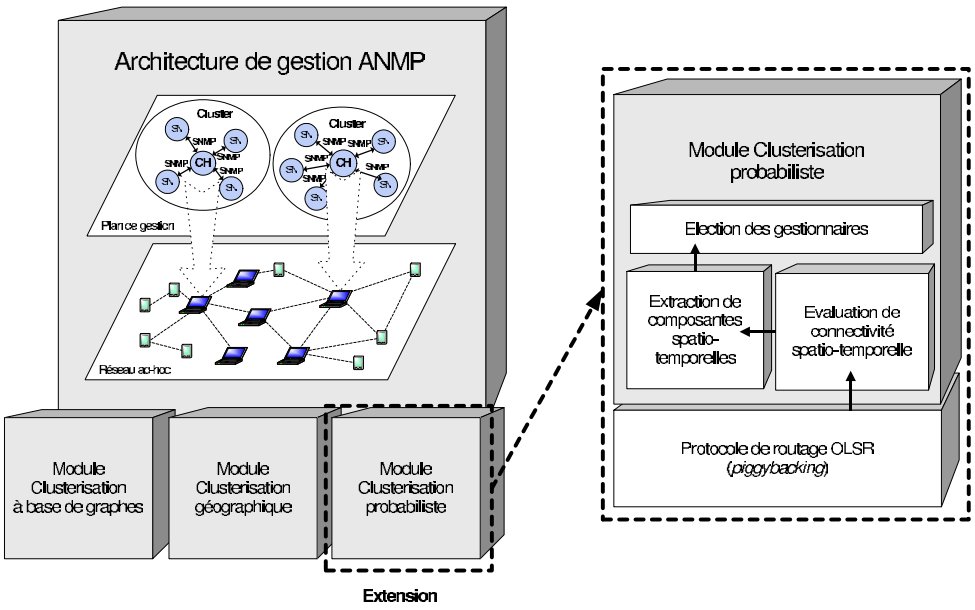


FIG. 5.4 – Intégration de l'approche probabiliste au sein de l'architecture ANMP [L'architecture ANMP est extensible à différents algorithmes de clusterisation. Afin d'organiser le plan de gestion sous la forme de composantes spatio-temporelles, nous introduisons un algorithme supplémentaire au-dessus du protocole de routage proactif OLSR.]

Ces algorithmes de clusterisation sont représentés par les deux premiers modules de la figure 5.4. L'architecture ANMP définit une distinction claire entre l'utilisation de clusters pour la gestion et l'utilisation de clusters pour le routage. Etant spécifié au niveau de la couche applicative, ANMP présuppose l'existence d'un protocole de routage dans les couches inférieures. La clusterisation dans le plan de gestion permet de construire logiquement le plan de gestion et sélectionner des nœuds intermédiaires comme gestionnaires locaux, afin de limiter hiérarchiquement le trafic de

gestion. Dans un algorithme de routage à base de clusters, l'objectif est d'assurer la maintenance des routes en donnant aux têtes de clusters la responsabilité de router les paquets en dehors d'un cluster donné. Ainsi, l'utilisation de clusters au niveau applicatif et au niveau routage ne s'inscrit pas dans la même finalité (organiser le plan de gestion et respectivement maintenir les routes). Cependant, les auteurs d'ANMP mettent en évidence dans [53] l'extensibilité de l'architecture à d'autres formes de clusterisation et notamment la possibilité de s'appuyer sur les informations de routage afin d'organiser le plan de gestion.

Nous proposons une extension de l'architecture ANMP à travers un module de clusterisation complémentaire qui implante notre méthode probabiliste de supervision. Il correspond au dernier module décrit à la figure 5.4. Ce module est organisé sous la forme de trois composants de base :

- le premier composant permet d'évaluer la connectivité spatio-temporelle des nœuds dans le réseau ad-hoc. Il s'appuiera autant que possible sur les informations fournies par le plan de routage afin de déterminer la probabilité, pour deux nœuds donnés, d'être dans le même voisinage et de pouvoir intercommuniquer.
- le second composant permet l'extraction des composantes spatio-temporelles. Il exploite les mesures de connectivité spatio-temporelle et implante l'algorithme 1 pour agréger les nœuds sous la forme de sous-ensembles disposant d'une forte connectivité spatio-temporelle : chaque nœud a été le voisin d'un autre nœud de la composante durant un pourcentage de temps minimal.
- le dernier composant assure l'élection des nœuds gestionnaires en considérant le mécanisme électif défini par l'algorithme 2. Les données relatives aux composantes spatio-temporelles sont directement fournies par le second composant. L'élection repose sur une évaluation de la centralité par vecteur propre des nœuds de la composante : le critère permet de ne pas se limiter à une mesure locale en prenant en compte de manière récursive l'importance relative des nœuds voisins.

Le module probabiliste est déployé au dessus du protocole de routage afin de réutiliser les données fournies dans ce plan et offrir une solution faiblement coûteuse en messages de contrôle.

5.4.2 Utilisation du protocole de routage comme support

L'évaluation de la connectivité spatio-temporelle repose sur une analyse du protocole de routage proactif standard OLSR (*Optimized Link State Routing Protocol*) [57] déjà présenté dans le chapitre 1. A l'instar d'un algorithme à état de liens classique, chaque nœud réalise deux opérations principales : (1) il détermine la liste des nœuds voisins directs en émettant périodiquement des paquets HELLO de beaconing et (2) il échange les informations d'états de liens avec les autres nœuds en diffusant des messages TC de contrôle de topologie dans le réseau. Notre méthode probabiliste s'intéresse à ces deux opérations de découverte du voisinage et de diffusion des informations de topologie.

Lors de l'émission des messages HELLO, un nœud OLSR utilise une base d'informations de voisinage (décrit dans la RFC 3626 section 4.3 [57]). Il maintient notamment la liste des nœuds voisins à un saut en enregistrant un ensemble de tuples de la forme (*N_neighbor_main_addr, N_status, N_willingness*) correspondant respectivement à l'adresse OLSR principale du voisin, l'état du lien (symétrique ou asymétrique) et un indicateur de sa volonté à participer à la tâche de routage. Nous définissons un champ supplémentaire pour chaque tuple, appelé *N_spatio_temporal_connectivity*, qui spécifie la connectivité spatio-temporelle, c'est-à-dire le pourcentage de temps moyen durant lequel le nœud a été voisin du nœud courant.

Le nouveau champ doit également être intégré aux messages de contrôle de topologie (messages TC définis dans la RFC 3626 section 9 [57]) qui effectuent l'annonce de l'état des liens,

de sorte que les informations sur la connectivité spatio-temporelle soient propagées dans le réseau ad-hoc. Le format d'un message TC décrit un ensemble de *Advertised Neighbor Main Address*. Nous introduisons un champ supplémentaire appelé *Advertised Neighbor Spatio Temporal Connectivity* pour annoncer le pourcentage du temps durant lequel deux nœuds ont été voisins.

A la réception des messages de topologie, chaque nœud ad-hoc construit la topologie du réseau en maintenant une liste contenant les informations sous la forme de tuples (*T_dest_addr, T_last_addr, T_seq*) dont les entrées correspondent respectivement à l'adresse principale de la destination, l'adresse principale du prochain saut et un numéro de séquence. Nous définissons également un champ supplémentaire *T_spatio_temporal_connectivity* pour prendre en compte la connectivité spatio-temporelle. Notre extension s'appuie sur une adaptation du protocole OLSR et fournit à ANMP un module supplémentaire lui permettant de fonctionner en mode probabiliste.

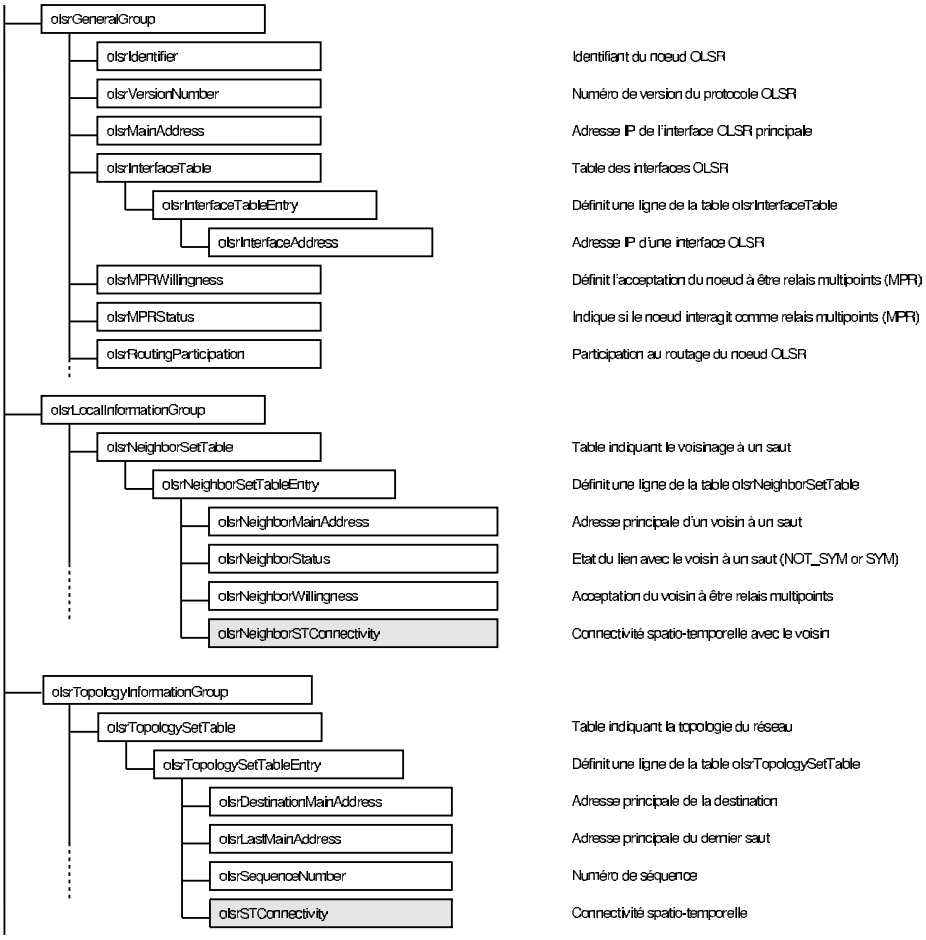


FIG. 5.5 – Extension de la MIB OLSR pour la gestion probabiliste [La figure présente une vue partielle de la MIB OLSR que nous avons proposée dans [13]. Les champs grisés correspondent à l'extension de la MIB permettant de prendre en charge la connectivité spatio-temporelle.]

5.4.3 Extension de la base d'informations

L'intégration de notre approche probabiliste sous-tend l'extension de la base d'informations de gestion (MIB) [159] de l'architecture ANMP et du protocole de routage OLSR. Nous avons défini précédemment une MIB OLSR dans [13] qui est partiellement décrite à la figure 5.5 et qui dérive de l'extension du modèle commun de l'information. Cette vue partielle décrit trois groupes distincts : le premier appelé *olsrGeneralGroup* fournit des informations d'ordre général sur le nœud OLSR telles que les interfaces réseau compatibles avec le protocole ou bien encore la volonté du nœud à participer au fonctionnement du plan de routage en tant que relais multipoints.

FIG. 5.6 – Extension de la MIB ANMP pour la gestion probabiliste [La figure détaille le groupe *anmpTopologyMaintenance* de cette MIB avec les entrées en gris correspondant à notre extension. Nous définissons un sous-groupe additionnel appelé *anmpProbaClustering* afin d'appliquer notre méthode probabiliste à l'organisation du plan de gestion.]

Le second groupe *olsrLocalInformationGroup* définit des informations sur le voisinage local

du nœud incluant la liste des voisins à un saut *olsrNeighborSetTable* et la liste des voisins à deux sauts *olsrTwoHopNeighborSetTable*. La table *olsrNeighborSetTable* énumère les voisins à un saut en précisant leur état de liens. Nous ajoutons à cette table une entrée supplémentaire *olsrNeighborSTConnectivity* (représentée en gris sur la figure). Cette entrée définit la connectivité spatio-temporelle avec le nœud voisin.

Le troisième groupe *olsrTopologyInformationGroup* définit les informations sur la topologie du réseau ad-hoc. Il inclut une table *olsrTopologySetTable* indiquant pour chaque adresse de destination l'adresse du dernier saut. Nous intégrons aussi dans cette table une nouvelle entrée notée *olsrSTConnectivity* pour maintenir les valeurs de connectivité spatio-temporelle.

La MIB de l'architecture ANMP est également adaptée en conséquence. Le groupe *anmpTopologyMaintenance* de cette MIB est détaillé à la figure 5.6 avec en gris les entrées correspondant à l'extension. Le groupe contient les informations relatives à la topologie du plan de gestion. La table *anmpNeighborTable* définit la liste des voisins à un saut qui est complétée par une entrée supplémentaire *anmpNeighborSTConnectivity* afin d'assurer la cohérence avec l'extension de la MIB OLSR. La table *anmpProtocolTable* maintient une entrée pour chaque algorithme qui peut être utilisé pour la maintenance de la topologie sous forme de clusters. Chacun des deux algorithmes de clusterisation fournis avec ANMP est représenté par un sous-groupe : *anmpGraphClustering* et *anmpGeographicalClustering*. Ces groupes ne sont pas détaillés sur la figure mais une description plus complète peut être trouvée dans [53].

Nous définissons un sous-groupe supplémentaire appelé *anmpProbaClustering* pour intégrer notre méthode probabiliste dans la base d'informations. Ce sous-groupe maintient les informations générées par l'algorithme d'extraction des composantes spatio-temporelles (algorithme 1) et par l'algorithme permettant l'élection des nœuds gestionnaires (algorithme 2) présentés dans la section précédente. En particulier, la table *anmpProbaSTComponentTable* référence la liste des composantes spatio-temporelles. Chaque entrée *anmpProbaSTComponentEntry* fournit les caractéristiques de la composante incluant la liste des nœuds gestionnaires détaillée dans la table *anmpProbaSTComponentManagerTable* et la liste des nœuds agents détaillée dans la table *anmpProbaSTComponentAgentTable*. Ces extensions de MIB nous permettent d'intégrer les informations de gestion induites par notre approche probabiliste au sein du modèle d'information.

5.5 Résultats expérimentaux

Nous détaillons dans cette section une évaluation des performances de notre approche de gestion. Nous décrivons différentes séries d'expérimentations dans lesquelles nous avons appliqué notre méthode algorithmique pour identifier les composantes spatio-temporelles les plus larges. Les expérimentations ont été réalisées à l'aide du simulateur réseau à événements discrets ns-2 [74] qui est largement utilisé dans la communauté et dispose de différentes extensions dédiées aux réseaux ad-hoc. Nous utilisons le protocole d'accès au médium correspondant à la norme IEEE 802.11 ainsi que le protocole de routage proactif OLSR dans le cadre des simulations. L'implantation du protocole de routage correspond à l'extension proposée par le NRL [5]. Nous avons considéré un réseau ad-hoc composé de $n \in [5 - 30]$ entités mobiles qui sont distribuées aléatoirement sur une surface de 1000 m x 1000 m.

Le déplacement des nœuds suit le modèle de mobilité *Random WayPoint* (RWP) [28, 22] qui reste un modèle de référence malgré ses limites [199, 35, 27]. Avec ce modèle, chaque nœud alterne des périodes de déplacement et des périodes d'immobilité. En période de déplacement, le point de destination d'un nœud est choisi de manière aléatoire sur la surface de simulation. La vitesse de déplacement vers ce point est constante et est choisie aléatoirement dans l'intervalle

borné [0 m/s − $vitesse\_max$ m/s]. Lorsque le nœud atteint la destination, il entre dans une période d'immobilité en effectuant une pause dont la durée est fixée en seconde par la variable *temps_pause*. Il opère ensuite un nouveau déplacement selon la même procédure vers une nouvelle destination et avec une autre vitesse. Dans nos simulations, la vitesse maximale peut varier de 0.5 m/s à 10 m/s et les temps de pause adoptés dans un scénario fluctuent de 1 s à 50 s. Les paramètres de simulation sont rappelés dans le tableau de synthèse 5.1. Ceux qui ne sont pas explicitement spécifiés correspondent aux valeurs par défaut du simulateur.

Paramètres	Valeurs
Simulateur	ns-2
Temps de simulation	1800 s
Surface de simulation	1000 m x 1000 m
Nombre de nœuds ad-hoc	5-30 nœuds
Modèle de mobilité	Random WayPoint *mobgen - steady state*
Vitesse	0.5 - 10 m/s
Temps de pause	1 - 50 s
Couche MAC	IEEE 802.11
Couche de routage	NRL OLSR

Tab. 5.1 – Paramètres de simulation

Pour éviter les problèmes d'initialisation avec le modèle de mobilité *Random WayPoint*, nous avons utilisé le générateur steady-state *mobgen-ss* [47] avec lequel les vitesses initiales et la position des nœuds sont directement sélectionnées à partir de la distribution stationnaire. En règle générale, la distribution initiale des vitesses et des positions diffère de celle observée à plus long terme lors d'une simulation. Elle varie de manière continue dans le temps jusqu'à converger vers la distribution steady-state ou distribution stationnaire. Pour limiter le biais induit par la distribution initiale sur les résultats de simulation, une méthode simple consiste à ne pas tenir compte des valeurs observées durant une période initiale. Mais cette méthode n'est pas efficace puisqu'il est difficile de fixer la durée de cette période initiale afin d'être certain d'atteindre la convergence. Avec une faible mobilité, la convergence peut prendre plus de 1000 s (soit plus de 16 min) de temps simulé. L'outil *mobgen-ss* permet de déterminer la distribution stationnaire pour le modèle de mobilité *Random WayPoint* et de sélectionner vitesses, temps de pause et positions des nœuds directement à partir de cette distribution, ceci dès l'initialisation.

5.5.1 Proportion des nœuds couverts par la première composante

Dans une première série d'expériences, nous souhaitions analyser la proportion des nœuds du réseau ad-hoc présents dans la première composante spatio-temporelle i.e. la composante contenant le plus grande nombre de nœuds du réseau. Nous avons réalisé un ensemble étendu de simulations avec différents paramètres de mobilité et différentes topologies de réseau et avons appliqué la méthode algorithmique de gestion. Pour chaque configuration individuelle, nous avons réalisé 150 simulations pour assurer le non biais des valeurs obtenues. Les résultats sont présentés à la figure 5.7 où nous avons tracé la probabilité de distribution de la proportion des nœuds présents dans la première composante.

L'axe des abscisses correspond au pourcentage minimum de nœuds qui font partie de la plus large composante spatio-temporelle, tandis que l'axe des ordonnées correspond au pourcentage de cas (simulations) où ce pourcentage a été mesuré. Ainsi, un point (x,y) sur le graphe représente le

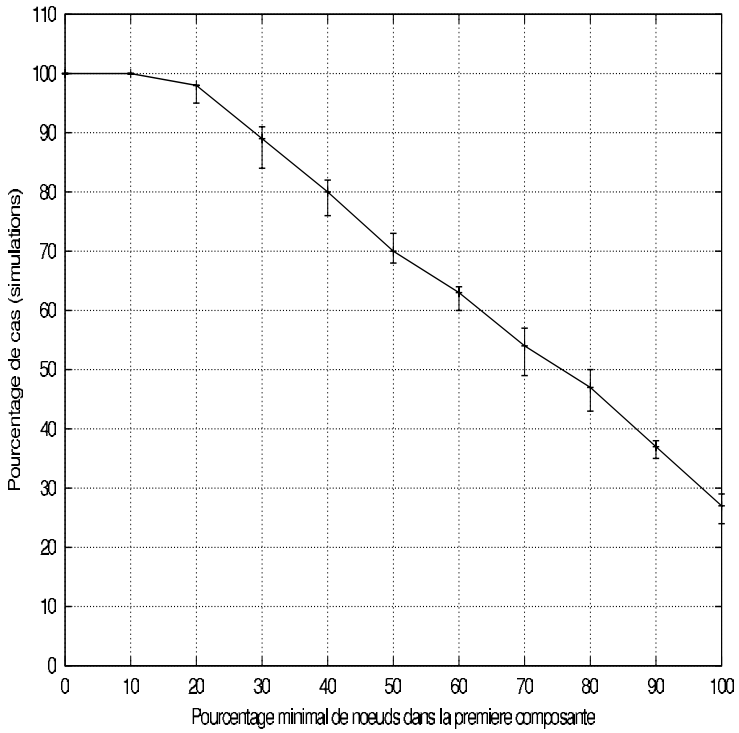


FIG. 5.7 – Probabilité de distribution du ratio de la première composante spatio-temporelle [Un point (x,y) sur le graphe indique le pourcentage y de simulations pour lesquelles la composante couvre au moins x pour-cent des nœuds du réseau.]

pourcentage de simulations (y) pour lesquelles la première composante couvrait au moins x pour-cent des nœuds ad-hoc. Par exemple, nous pouvons observer que la probabilité d'avoir au moins 25% des nœuds dans la première composante est d'environ 95%. A l'inverse, la probabilité d'avoir au moins 60% des nœuds dans la composante est d'environ 63% dans le cadre de nos simulations. Ainsi, le pourcentage de nœuds dans la première composante décroit avec le pourcentage de cas avérés, ce qui se manifeste par une courbe décroissante.

Des garanties quant au pourcentage de nœuds gérés (i.e. présents dans la première composante) peuvent être inférées à partir de cette distribution de probabilité. Nous pourrions ainsi considérer une approche pragmatique fondée sur une gestion *best effort* de la forme : nous gérons environ 25% des nœuds et dans ce cas nous garantissons que cette contrainte sera respectée dans 95% des cas. Nous pouvons de cette façon assurer une gestion de la moitié des nœuds ad-hoc avec une probabilité proche de 70%. Cette approche *best effort* pourrait être prise en charge par le nœud gestionnaire de la composante.

5.5.2 Impact du modèle de mobilité sur la méthode

Une question naturelle consiste à se demander dans quelle mesure la mobilité du réseau a un impact sur notre organisation du plan de gestion. Intuitivement, une mobilité plus élevée devrait dégrader les performances du système en générant des composantes de taille plus réduite pour une même valeur de probabilité. Ainsi, la mobilité des nœuds diminuerait le pourcentage de temps durant lequel deux nœuds ont été dans le même voisinage. Mais, une quantification précise du phénomène est nécessaire pour en évaluer l'impact sur la construction des composantes. Nous

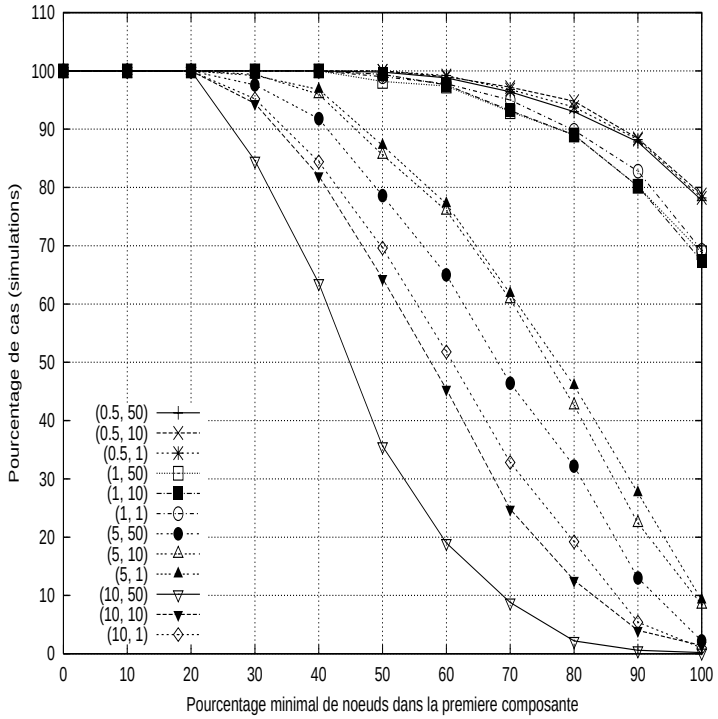


FIG. 5.8 – Probabilité de distribution du ratio de la première composante spatio-temporelle en fonction de la mobilité du réseau [Chaque courbe représente des paramètres de mobilité différents $(vitesse\_max, temps\_pause)$.]

avons donc réalisé une seconde série d'expériences qui porte spécifiquement sur ce problème en analysant l'influence des différents paramètres de mobilité. Les résultats sont détaillés à la figure 5.8 où chaque courbe représente la probabilité de distribution pour un couple de paramètres de mobilité différents $(vitesse\_max, temps\_pause)$. Nous avons réalisé 150 simulations pour chaque configuration pour éviter une fois encore tout phénomène de biais.

Nous nous attendions à des résultats performants pour une mobilité faible (c'est-à-dire une vitesse maximale faible et un temps de pause élevé), où par résultats performants nous entendions une forte probabilité d'avoir une première composante contenant un grand nombre de nœuds dans le réseau. Si nous observons le cas d'une vitesse de 0.5 m/s et d'un temps de pause de 50 s, nous constatons que c'est en effet le cas avec environ 90% des nœuds situés dans la composante principale avec une probabilité proche de 90%.

Cependant, un autre constat, surprenant celui-là, est obtenu lorsque nous analysons les différents résultats en variant le temps de pause mais à vitesse constante $vitesse\_max = 10$ m/s. Les pires résultats sont obtenus avec un temps de pause élevé dans cette configuration. Ceci contredit notre hypothèse initiale qu'une forte mobilité induit nécessairement des composantes principales de taille plus faible. Il semble que dans les cas où les nœuds ont une vitesse élevée, la connectivité spatio-temporelle est améliorée lorsque les nœuds effectuent de court temps de pause. Ainsi, un tel résultat conduit à la condition suivante : si le nœud a une vitesse de déplacement élevée, il est préférable qu'il effectue des pauses de faible durée.

5.5.3 Importance relative de la seconde composante

Dans les expériences précédentes, nous nous sommes limités à l'analyse de la plus large composante spatio-temporelle. Une extension possible de ce travail est de considérer les deux plus importantes composantes. La figure 5.9 représente les résultats obtenus à la suite d'une série d'expériences identique à celle décrite à la figure 5.7, mais où nous agrégeons cette fois-ci les tailles des deux premières composantes. Nous pouvons observer en conséquence des performances

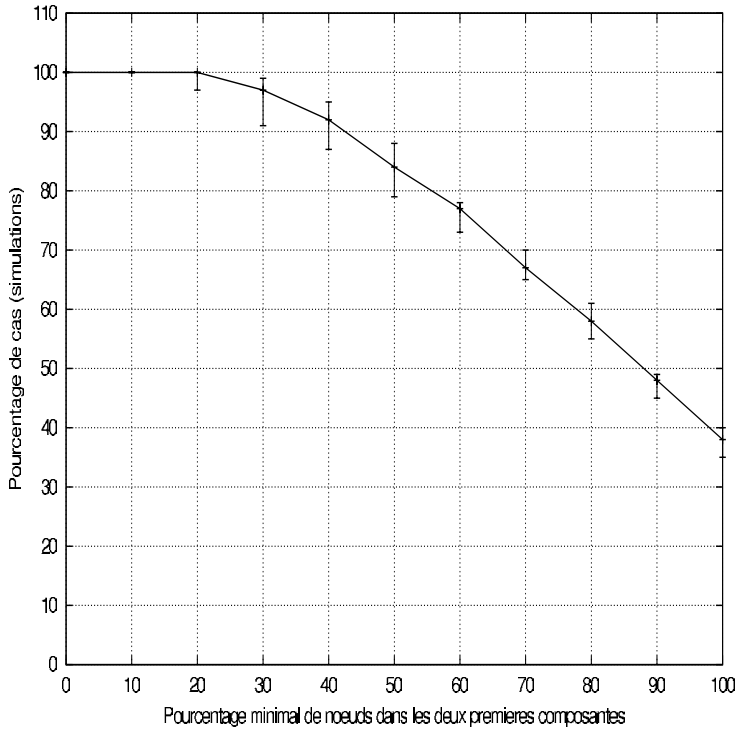


FIG. 5.9 – Probabilité de distribution du ratio des deux premières composantes spatio-temporelles [La courbe représente la valeur agrégée des probabilités de distribution des deux composantes les plus larges.]

améliorées avec un pourcentage minimal de nœuds gérés plus élevé pour une même valeur de probabilité. Ainsi, environ 50% des nœuds sont situés dans les deux premières composantes avec une probabilité proche de 85%, contre une probabilité de 70% auparavant dans le cas d'une unique composante. Cela signifie que l'on peut gérer environ 50% des nœuds avec une très forte probabilité dans les conditions énoncées. Globalement, nous constatons, par comparaison avec la courbe de la figure 5.7 que nous obtenons environ 15% de nœuds gérés en plus avec une même probabilité. Si nous considérons une approche *best effort* avec la contrainte de gérer au moins 50% des nœuds, alors nous pouvons atteindre ces objectifs dans 85% des cas. Nous avons détaillé à la figure 5.10 la taille individuelle des deux premières composantes (exprimée en pourcentage de nœuds couverts dans le réseau). Nous observons dans ce cas que la seconde composante la plus large est significative dans environ 80% des cas et compte pour 15% de la taille du réseau (la seconde composante étant par définition de taille inférieure ou égale à la première composante).

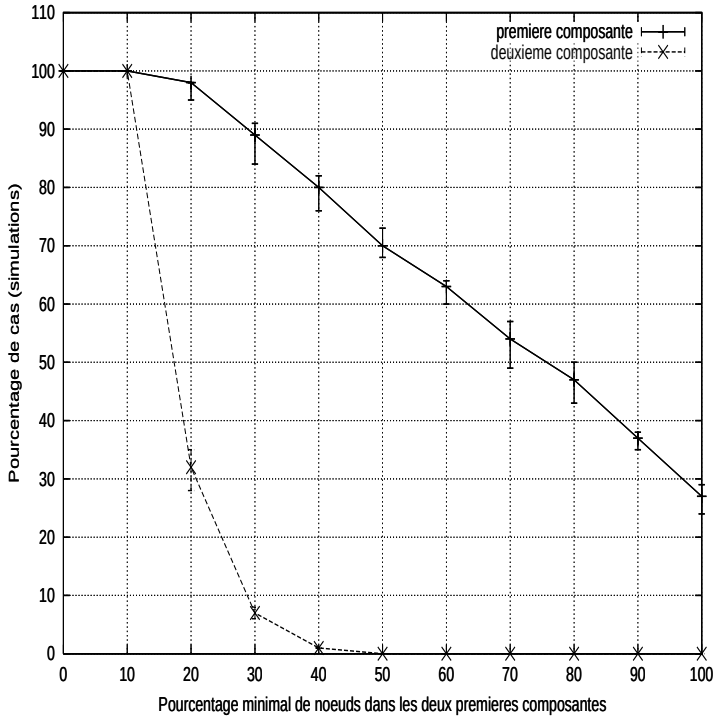


FIG. 5.10 – Probabilité de distribution du ratio de chacune des composantes spatio-temporelles

5.5.4 Comparaison avec un modèle de mobilité de groupes

Dans une dernière série d'expériences, nous souhaitions évaluer si une mobilité alternative [46], en particulier une mobilité de groupes, permettait d'améliorer les performances de notre méthode probabiliste. Dans le modèle RWP (*Random WayPoint*), les déplacements sont définis de manière complètement indépendante pour chacun des nœuds. Le modèle RPGM *(Reference Point Group Mobility*) [110] permet de simuler une mobilité de groupes où les déplacements d'un groupe sont déterminés par un point de référence logique. Chaque nœud du groupe suit ce point de référence lors de ses déplacements avec une déviation donnée.

La figure 5.11 décrit les résultats obtenus en expérimentant le modèle RPGM avec différentes tailles de groupe (25%, 50% et 75% exprimés relativement à la taille du réseau). Nous constatons que les pourcentages de nœuds présents dans la première composante offrent dans les trois situations de meilleurs résultats qu'avec le modèle RWP. Ainsi, la gestion d'au moins la moitié des nœuds est possible avec une probabilité de 65% à 80% en fonction de la taille des groupes. Mais, l'amélioration est encore plus caractéristique lorsque la contrainte sur le pourcentage de nœuds à gérer est élevée. Ainsi, si nous souhaitons gérer au moins 90% des nœuds dans le réseau, la probabilité passe d'environ 35% avec le modèle RWP à une probabilité de 55% avec le modèle RPGM (taille de groupe de 50%). Une telle mobilité permet en effet de faciliter la construction des composantes en définissant des déplacements davantage contraints.

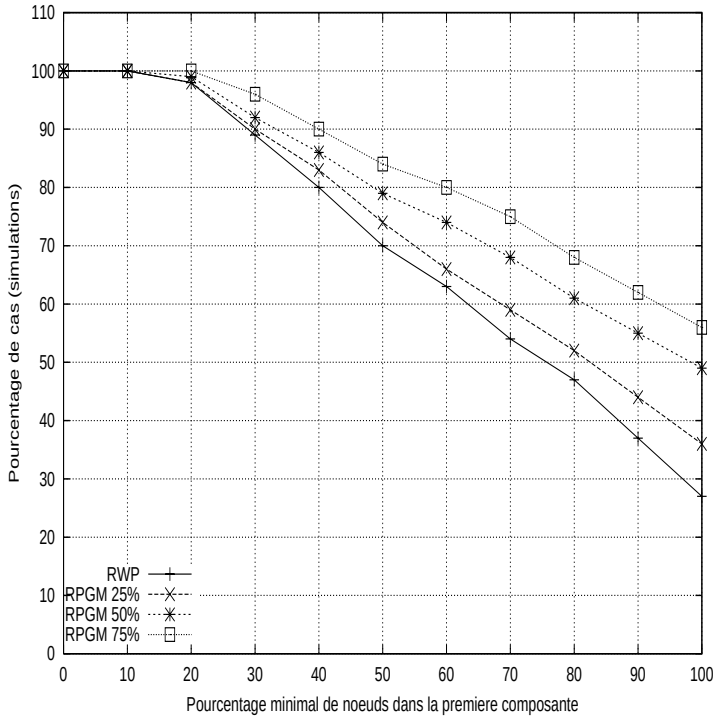


FIG. 5.11 – Probabilité de distribution du ratio de la première composante avec le modèle de mobilité de groupes RPGM [La courbe obtenue avec le modèle RWP est comparée avec les courbes issues du modèle RPGM avec différentes tailles de groupe (exprimées en pourcentage relativement à la taille du réseau).]

5.6 Synthèse

Au lieu de considérer une approche de gestion au sens pur du terme où l'ensemble des nœuds serait géré, nous avons présenté dans ce chapitre une méthode probabiliste pour configurer le plan de gestion de manière plus souple. L'idée maîtresse consiste à restreindre volontairement le nombre de nœuds gérés à travers la construction de composantes spatio-temporelles. Une telle composante correspond à un sous-ensemble de nœuds dont les éléments ont une forte probabilité d'être directement connectés dans le temps. Le terme spatial est dérivé de la notion de voisinage local tandis que le terme temporel signifie que nous ne nous limitons pas à une vue instantanée. Nous ne cherchons pas à garantir que les nœuds soient connectés en permanence sur une période de temps mais considérons une démarche moins stricte où nous identifions les nœuds qui ont été voisins durant un pourcentage de temps minimum. Le plan de gestion est volontairement limité aux composantes spatio-temporelles les plus larges afin de limiter l'impact de la supervision sur le fonctionnement même du réseau.

Nous avons décrit la méthode algorithmique correspondante, qui permet à la fois d'extraire les composantes spatio-temporelles et de sélectionner les nœuds gestionnaires. En particulier, nous avons détaillé divers mécanismes électifs fondés sur la centralité de degré et la centralité par vecteur propre et montré comment les appliquer sur des scénarios concrets. Ces mécanismes permettent de favoriser les nœuds qui ont une importance structurelle dans une composante afin de garantir que la tâche de gestion soit davantage robuste. La méthode algorithmique a été

intégrée à l'architecture de gestion ANMP en exploitant les informations du plan de routage fournies par le protocole OLSR.

Notre approche est qualifiée de *probabiliste* car son fonctionnement ne peut être garanti que de manière stochastique : nous assurons qu'un pourcentage donné des nœuds du réseau sera géré avec une certaine probabilité. Nous avons estimé les distributions de probabilité à travers un ensemble étendu de simulations. Nous avons notamment montré qu'avec le modèle de mobilité *Random WayPoint* (RWP) la moitié des nœuds peut être gérée avec une probabilité de l'ordre de 70% dans le cadre considéré. Une analyse plus fine a permis de quantifier l'impact de la mobilitié avec le modèle RWP ainsi qu'avec le modèle de mobilité de groupes RPGM. On constate en particulier qu'une mobilité de groupes facilite la construction des composantes à travers des déplacements plus contraints. Bien que la simulation présente des limites fonctionnelles [36], la méthodologie expérimentale reste générique et peut être facilement mise en œuvre avec d'autres modèles de mobilité alternatifs. Notre démarche probabiliste permet de mieux configurer et gérer le plan de gestion en le limitant volontairement à un certain pourcentage de nœuds. Mais, indépendamment de l'aspect organisationnel, les opérations de gestion ne doivent-elles pas, elles aussi, être adaptées aux contraintes des réseaux ad-hoc ?

Les travaux présentés dans ce chapitre ont fait l'objet de plusieurs publications [19, 15, 21].

Chapitre 6

Gestion de performances par filtrage et analyse de graphes

Sommaire

6.1 Introduction

La gestion de performances est la première aire fonctionnelle pour laquelle nous souhaitions adapter les opérations de gestion. Elle consiste à évaluer l'état de fonctionnement du réseau et à maintenir par des opérations correctives une qualité de service [9, 10]. Nous nous sommes intéressés à définir des méthodes d'analyse permettant de construire une vue fonctionnelle synthétique du réseau ad-hoc et de déterminer l'impact des nœuds sur le fonctionnement du réseau.

Dans une infrastructure fixe traditionnelle, la topologie du réseau évolue sur des échelles de temps larges car une modification implique en règle générale des interventions physiques : déploiement d'un nouveau routeur, installation de nouveaux câblages. D'autre part, les entités du réseau, notamment les nœuds d'interconnexion, ont des fonctionnalités spécifiques : les équipements routeurs sont clairement identifiés lors du déploiement du réseau et il est possible pour un administrateur de déterminer et de quantifier les points du réseau où l'activité sera la plus

importante. Dans un réseau ad-hoc, la topologie est fortement dynamique [23] car l'organisation du réseau s'opère de manière autonome. Le comportement des nœuds n'est pas connu *a priori* puisque chaque entité mobile peut à la fois intervenir comme terminal et comme nœud d'interconnexion.

Il est nécessaire de mettre en œuvre une stratégie qui permette tout d'abord de faire apparaître les *backbones* du réseau ad-hoc. Nous entendons par *backbone* les chemins qui sont les plus utilisés par les nœuds ad-hoc lors de communications multi-sauts. D'autre part, il convient de déterminer l'impact des nœuds sur le fonctionnement global du réseau en mettant en évidence les disparités entre nœuds. Cet impact peut être positif (typiquement, les nœuds qui ont une activité de routage importante et qui font partie d'un *backbone*) ou négatif (typiquement, les nœuds qui refusent d'intervenir comme routeurs ou qui consomment abusivement les ressources du réseau). Les données issues de cette observation peuvent être exploitées à des fins de :

- reconfiguration du réseau : une connaissance de l'état fonctionnel du réseau permet d'adapter les paramètres de configuration, en l'occurence ceux portant sur les nœuds ad-hoc, afin d'aboutir à un fonctionnement optimal en fonction des propriétés du réseau. Ainsi dans l'architecture de gestion GUERRILLA [177] présentée dans le chapitre 1, nous constations que le mécanisme d'auto-configuration reposait sur une évaluation de l'état du réseau et sur la prédiction de son état futur par rapport à une éventuelle opération de reconfiguration.
- positionnement de sondes : l'identification de nœuds ad-hoc qui ont une forte activité de routage permet d'automatiser le positionnement de sondes. Ce placement doit bien évidemment être multi-critères et tenir compte également des capacités de traitement et de la durée de vie potentielle du nœud. A l'inverse, il faut veiller à ce que l'instrumentation des nœuds actifs ne provoque pas une dégradation de la performance.
- provisionnement de nœuds : un opérateur peut identifier les besoins des usagers à travers cette évaluation. En particulier, il peut permettre, dans le cas de réseaux maillés sans fil, la migration ou le déploiement de nœuds supplémentaires dans une zone de forte densité de trafic.

Ce chapitre sera organisé de la manière suivante : nous commencerons par présenter deux métriques de performance qui permettent de caractériser le fonctionnement d'un nœud ad-hoc. Ensuite, nous détaillerons des méthodes d'analyse, reposant sur une technique de filtrage et sur une analyse des graphes de dépendances, qui seront appliquées à partir de ces métriques. Le filtrage permet de comparer l'état des nœuds dans un voisinage local et l'analyse de graphes met en évidence l'impact des nœuds en fonction de leurs dépendances avec les autres nœuds. Nous décrirons enfin un ensemble de résultats expérimentaux obtenus en appliquant ces méthodes d'analyse à différents scénarios.

6.2 Métriques de performances

Une métrique de performances est un critère de mesure permettant de quantifier les performances d'un système [4]. Nous introduisons deux métriques de performances afin de quantifier l'état fonctionnel d'un nœud ad-hoc. Nous souhaitions mettre en évidence la nature bivalente d'un tel nœud : un nœud ad-hoc intervient à la fois comme terminal lorsqu'il émet ou reçoit des paquets pour son propre compte, et à la fois comme routeur lorsqu'il émet et reçoit des paquets pour le compte d'autres nœuds. La première métrique appelée degré d'atteignabilité traduit la capacité d'un nœud en tant que terminal à communiquer avec les autres nœuds ad-hoc, et la seconde métrique exprime la participation du nœud en tant que routeur au fonctionnement du réseau.

6.2.1 Degré d'atteignabilité

Un des objectifs premiers d'un réseau ad-hoc est de permettre à un ensemble de nœuds mobiles de communiquer entre eux, et ce sans nécessairement utiliser une infrastructure fixe. Nous proposons une première métrique définie au niveau applicatif, appelée degré d'atteignabilité, permettant d'exprimer la capacité d'un nœud à communiquer avec les autres nœuds et nous la définissons ci-dessous. En considérant la modélisation suivante : soit un réseau ad-hoc correspondant à un ensemble $V = \{v_1, v_2, v_3 \dots v_n\}$ de n nœuds mobiles se déplaçant sur un terrain donné, alors le degré d'atteignabilité du nœud v_i à l'instant t, noté $DA(v_i, t)$, est défini comme la part des nœuds du réseau ad-hoc que v_i peut atteindre à l'instant t, soit mathématiquement :

$$DA(v_i, t) = \frac{|\{v_j \ \in V |\ j \neq i \ \wedge \ v_i \ atteint \ v_j \ à \ l'instant \ t\}|}{n-1} \tag{6.1}$$

La condition v_i *atteint* v_j est vraie si et seulement si le nœud v_i est capable de communiquer avec le nœud v_j au niveau applicatif. Cette métrique normée tend vers 1 lorsque le nœud peut atteindre l'ensemble des nœuds du réseau. Elle peut être moyennée pour connaître le degré d'atteignabilité d'un nœud dans le temps. Ainsi, si la période $T = [0, t] \subset \Re^+$ correspond à la durée depuis laquelle le réseau ad-hoc a été déployé, alors le degré d'atteignabilité d'un nœud ad-hoc v_i est donné par la première égalité de l'équation 6.2.

$$DA(v_i) = \frac{1}{T} \int_0^T DA(v_i, t)\, dt \approx \frac{1}{k} \sum_{p=0}^{k-1} DA(v_i, t_p) \tag{6.2}$$

Afin de monitorer cette métrique, nous considérons une approche active [7] où des paquets de contrôle (assimilables à de simples pings) sont transmis périodiquement durant la période T pour évaluer $DA(v_i, t)$. T est divisée en k intervalles de temps égaux et les mesures sont réalisées périodiquement aux instants $T_{mesures} = \{t_0, t_1, t_2, t_3 \dots t_{k-1}\}$ avec $t_p = p \times \frac{T}{k}$. Le DA de v_i est alors obtenu classiquement par approximation comme décrit dans la seconde égalité de l'équation 2.

Un monitorage actif [126] repose sur l'injection de paquets spécifiques au sein du réseau ad-hoc, et sur l'observation de la manière avec laquelle ces paquets sont traités par les nœuds pour déterminer le niveau de service offert par le réseau. Ceci induit une charge supplémentaire qui doit être évaluée : notamment si le degré d'atteignabilité est mesuré dans le cadre d'un réseau ad-hoc à large échelle. Nous considérons en particulier que le degré d'atteignabilité d'un nœud ad-hoc peut se limiter à un sous-ensemble de nœuds. Il peut être davantage réaliste de s'intéresser uniquement à la capacité d'un nœud à atteindre une liste de nœuds ad-hoc favoris. Dans ce cas de figure, nous intégrons à notre métrique un paramètre supplémentaire, l'ensemble $S \subset V$, pour spécifier le sous-ensemble de nœuds, ce qui aboutit aux équations 6.3 et 6.4.

$$DA(v_i, S, t) = \frac{|\{v_j \ \in S |\ j \neq i \ \wedge \ v_i \ atteint \ v_j \in V \ à \ l'instant \ t\}|}{m-1} \tag{6.3}$$

$$DA(v_i, S) = \frac{1}{T} \int_0^T DA(v_i, S, t)\, dt \approx \frac{1}{k} \sum_{p=0}^{k-1} DA(v_i, S, t_p) \tag{6.4}$$

Nous nous limitons cette fois-ci à l'évaluation de la condition v_i *atteint* v_j pour les nœuds v_j faisant partie du sous-ensemble favori S. La mesure est définie au niveau applicatif dans le but d'offrir une évaluation concrète de l'état opérationnel d'un nœud. Ainsi, elle complète les informations qui pourraient être fournies par le plan de routage, en particulier les données de la table de routage lorsque le protocole est de nature proactive.

6.2.2 Participation au routage

Dans un réseau ad-hoc, la tâche de routage est réalisée par les entités mobiles elles-mêmes. Aussi, nous introduisons une seconde métrique, notée $PR(v_i)$, pour caractériser la participation au routage d'un nœud ad-hoc. Contrairement à la métrique précédente, elle est obtenue par un monitorage passif, c'est-à-dire sans injecter un trafic spécifique mais en analysant le trafic réseau courant niveau IP. La figure 6.1 présente le trafic réseau d'un nœud ad-hoc en le décomposant comme suit pour un nœud donné v_i : (1) $PRP(v_i)$ nombre de paquets reçus propre au nœud, (2) $PRR(v_i)$ nombre de paquets reçus pour le routage (dont le destinataire final n'est pas le nœud), (3) $PEP(v_i)$ nombre de paquets émis propre au nœud (dont il est l'émetteur initial), (4) $PER(v_i)$ nombre de paquets émis pour le routage et (5) $Pj(v_i)$ paquets jetés. Ainsi, le trafic est caractérisé par un double critère selon qu'il est entrant ou sortant et selon qu'il est manipulé par le nœud en tant que terminal ou en tant que routeur.

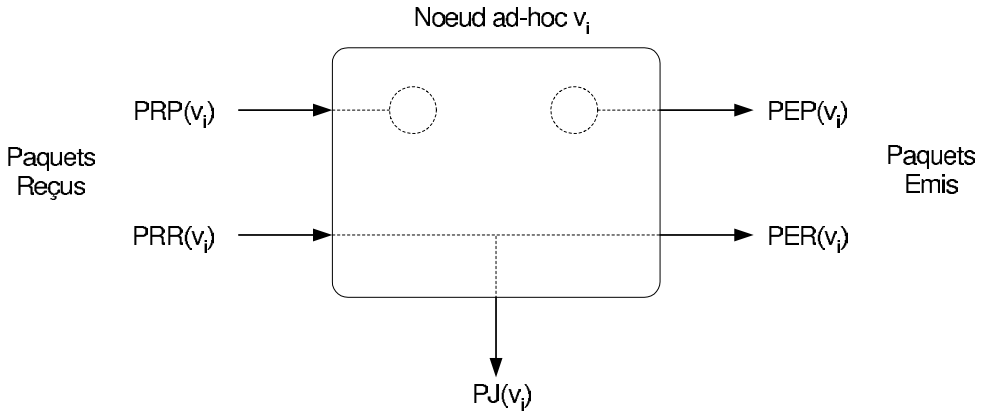


FIG. 6.1 – Schéma récapitulatif des trafics entrant et sortant d'un nœud ad-hoc

La métrique $PR(v_i)$ est obtenue à partir des mesures $PRR(v_i)$, $PER(v_i)$ et $PEP(v_i)$, comme décrit dans l'équation 6.5. Cette équation se décompose en deux membres distincts qui traduisent respectivement un élément de perte et un élément de charge.

$$PR(v_i) \quad = \quad (1 - \frac{PRR(v_i) - PER(v_i)}{PRR(v_i)}) \times (\frac{PER(v_i)}{PEP(v_i) + PER(v_i)}) \tag{6.5}$$

Le premier membre de l'équation rend compte de la perte de paquets à router au sein du nœud, obtenue en comparant les paquets reçus $PRR(v_i)$ et les paquets émis $PER(v_i)$ à router. Un nœud participe d'autant plus à la tâche de routage qu'il a été capable de retransmettre un grand nombre de paquets à router, i.e. le nombre de paquets jetés est faible. Mais la métrique doit également prendre en considération la capacité et le contexte du nœud. Si des perturbations physiques empêchent le nœud de retransmettre des paquets, une faible retransmission ne signifie pas nécessairement qu'il refuse de participer au routage. Le second membre exprime la part du trafic routé $PER(v_i)$ par le nœud relativement au trafic total $(PEP(v_i) + PER(v_i))$. Ainsi, une faible retransmission n'est pas considérée comme pathologique si elle est combinée à une faible émission de trafic. La métrique $PR(v_i)$ ne fournit pas une valeur discrète, mesurée à un instant t, mais est monitorée, selon une approche passive [126], à partir des statistiques sur le nombre de paquets cumulé sur une période T. Elle permet d'évaluer l'impact du nœud en tant que routeur sur le fonctionnement du réseau ad-hoc et d'identifier les nœuds qui ont une forte participation au routage.

6.3 Méthodes d'analyse des mesures de performance

Á partir des métriques de performances que nous avons définies dans la section précédente, nous proposons deux méthodes d'analyse présentées à la figure 6.2 permettant d'extraire une vue synthétique du réseau. La première approche définit une analyse à base de filtres dont l'objectif est de mettre en évidence des patrons de trafics et de comportements. La seconde repose sur une analyse plus fine fondée sur des graphes de dépendances afin d'identifier les nœuds importants. Les deux méthodes d'analyse sont complémentaires : la méthode d'analyse de

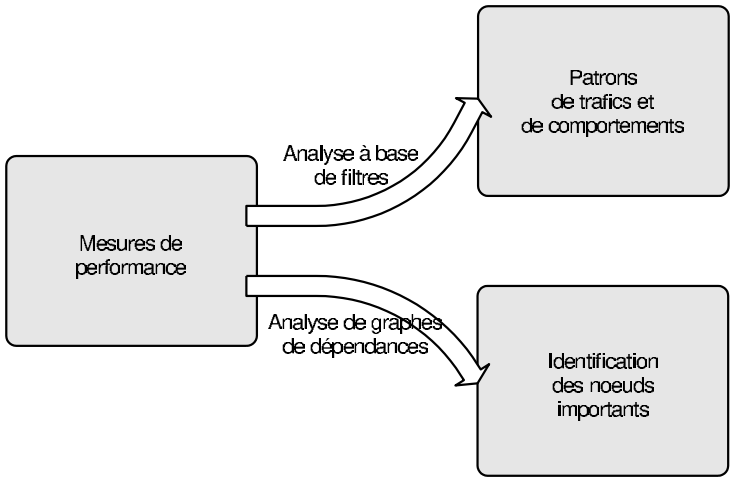


FIG. 6.2 – Méthodes d'analyse des mesures de performance

graphes de dépendances vise essentiellement à automatiser l'identification de nœuds importants qui ont été mis en évidence lors de l'analyse à base de filtres.

6.3.1 Analyse à base de filtres

L'analyse à base de filtres [102] consiste à traiter les mesures de performances à l'aide d'un filtrage de différence ou de similitude pour aboutir à une synthèse de l'état fonctionnel du réseau ad-hoc. Le fonctionnement des filtres considérés permet de comparer les mesures de performances d'un nœud donné relativement à celle de son voisinage. Si nous considérons une métrique de performances notée m, alors chaque nœud $v_i \in V$ effectue une mesure notée $m(v_i)$ de cette métrique et la compare à la mesure de ces voisins. Nous introduisons deux filtres : un filtre f_1 de différence et un filtre f_2 de similitude pour mettre en évidence les nœuds qui ont respectivement des propriétés différentes ou similaires.

Chacun des filtres s'applique en suivant deux étapes, mais la seconde étape est identique dans les deux cas. La première étape est une étape de comparaison : la mesure d'un nœud local est comparée aux mesures du voisinage à un saut, en suivant l'équation 6.6 pour le filtre f_1 ou l'équation 6.7 pour le filtre f_2. Dans le premier cas, le filtrage consiste à soustraire la valeur moyenne du voisinage avec la valeur courante du nœud afin de faire apparaître des disparités. Ainsi, si les nœuds voisins de v_i ont une valeur de métrique importante, alors il diminue l'influence du nœud v_i. Dans le second cas, le filtrage consiste à additionner la valeur moyenne du voisinage avec la valeur courante du nœud afin de faire apparaître des similitudes. Ainsi, si les nœuds voisins de v_i ont une valeur de métrique importante, alors ils contribuent à augmenter l'influence du

nœud v_i.

$$f_1(m(v_i)) = m(v_i) - \frac{\sum_{v_j \in\ N(v_i)} m(v_j)}{|N(v_i)|} \tag{6.6}$$

$$f_2(m(v_i)) = m(v_i) + \frac{\sum_{v_j \in\ N(v_i)} m(v_j)}{|N(v_i)|} \tag{6.7}$$

Sur ces équations, v_i correspond au nœud courant, $m(v_i)$ représente la valeur de la métrique appliquée au nœud v_i et $N(v_i)$ est l'ensemble des voisins à un saut de v_i. Si nous détaillons le fonctionnement des filtres sur un scénario simple avec quatre nœuds $\{v_1, v_2, v_3, v_4\}$, comme présenté à la figure 6.3. L'application du filtre f_1 au nœud v_2 revient à imputer la valeur moyenne de v_1, v_3 et v_4, tel que défini dans l'équation 6.8. Ainsi, si les mesures $\{m(v_1), m(v_2), m(v_3), m(v_4)\}$ sont

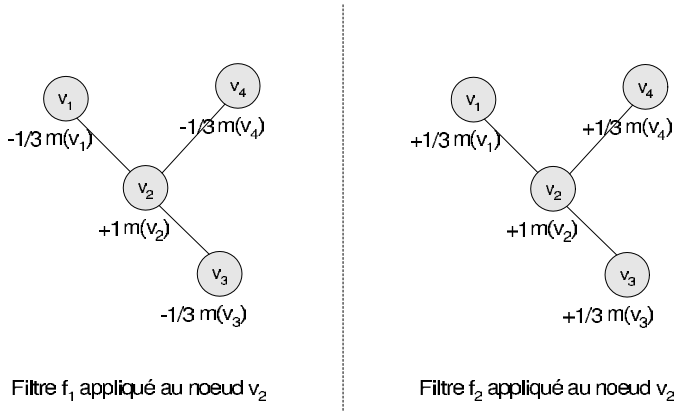


FIG. 6.3 – Application des filtres de différence f_1 et de similitude f_2

respectivement de {15, 10, 15, 15}, alors la valeur après filtrage $f_1(m(v_2))$ sera négative et vaudra -5. Inversement, si les valeurs des voisins sont inférieures à celle de v_2, $\{m(v_1), m(v_2), m(v_3), m(v_4)\}$ correspondant par exemple à {5, 10, 5, 5}, alors la valeur après filtrage $f_1(m(v_2))$ sera positive et vaudra $+5$. Ainsi, le filtre f_1 considère que la présence de voisins faibles renforce l'importance du nœud tandis que la présence de voisins forts la restreint.

$$f_1(m(v_2)) = m(v_2) - \frac{m(v_1) + m(v_3) + m(v_4)}{3} \tag{6.8}$$

Si l'on considère désormais le filtre f_2 de similitude défini par l'équation 6.9, la valeur moyenne des voisins est alors ajoutée à celle du nœud courant v_2. Si les mesures $\{m(v_1), m(v_2), m(v_3), m(v_4)\}$ sont respectivement de {15, 10, 15, 15}, alors la valeur après filtrage $f_2(m(v_2))$ passera à 25. Inversement, si les valeurs des voisins sont inférieures à celle de v_2, $\{m(v_1), m(v_2), m(v_3), m(v_4)\}$ correspondant à {5, 10, 5, 5}, alors la valeur après filtrage $f_2(m(v_2))$ vaudra 15.

$$f_2(m(v_2)) = m(v_2) + \frac{m(v_1) + m(v_3) + m(v_4)}{3} \tag{6.9}$$

Ainsi, la stratégie est à l'opposé de ce qui a été spécifié avec le filtre f_1 : la présence de voisins forts contribue à renforcer l'importance du nœud, tandis que la présence de voisins faibles la dégrade. La seconde étape du filtrage correspond à une opération de seuillage qui est définie par l'équation 6.10 :

$$f_s(m(v_i)) = \begin{cases} 0 & \text{if} & & & f_x(m(v_i)) & \leq & \varepsilon_1 \\ 1 & \text{if} & \varepsilon_1 & < & f_x(m(v_i)) & \leq & \varepsilon_2 \\ 2 & \text{if} & \varepsilon_2 & < & f_x(m(v_i)) & & \end{cases} \tag{6.10}$$

avec $x \in \{1, 2\}$ correspondant à l'indice du filtre et $\{\varepsilon_1 , \varepsilon_2\}$ étant des valeurs seuils configurables. Bien que le calcul du filtre puisse être réalisé de manière centralisée par un gestionnaire, les opérations ne requièrent pas une connaissance complète du réseau. Nous préférerons donc une approche où le calcul est réalisé localement par chacun des nœuds de manière distribuée, pour des raisons de performances et de passages à l'échelle. Une utilisation concrète des filtres peut être considérée pour l'analyse de la participation au routage et la mise en évidence de *backbones* de trafic dans le réseau ad-hoc.

6.3.2 Analyse de graphes de dépendances

La méthode de filtrage permet de mettre en évidence les disparités et les similitudes des éléments d'un réseau ad-hoc et de faire apparaître des patrons de trafics et de comportements. Nous proposons une approche plus fine fondée sur l'analyse d'un graphe de dépendances [151] permettant d'automatiser la détection de nœuds importants dans la structure du réseau. Nous construisons un graphe de dépendances orienté à partir des mesures effectuées au sein du réseau, en partant du postulat qu'un nœud est d'autant plus important qu'il est connecté à un nœud important. Ainsi, un nœud ayant une forte activité de routage a de grande chance de faire partie d'un *backbone* s'il est lui-même connecté à des nœuds s'impliquant fortement dans la tâche de routage.

Construction du graphe

Le graphe de dépendances est représenté classiquement sous la forme d'une matrice d'adjacence A : une ligne/colonne représente un nœud du réseau et une entrée $A[i, j]$ traduit un arc direct d'un nœud v_i à un nœud v_j. Un arc de v_i à v_j représente la dépendance du nœud v_i envers le nœud v_j et réciproquement l'influence du nœud v_j sur le nœud v_i. Nous considérons qu'un nœud a une influence sur un autre nœud voisin s'il dispose d'une mesure plus performante que celui-ci, ce qui se traduit mathématiquement par l'équation 6.11 :

Pour chaque nœud $v_i \in V$, pour chaque nœud $v_j \in N(v_i)$,

$$A(i,j) = \begin{cases} 1 & \text{si } m(v_i) > m(v_j) \\ 0 & \text{dans le cas contraire.} \end{cases} \tag{6.11}$$

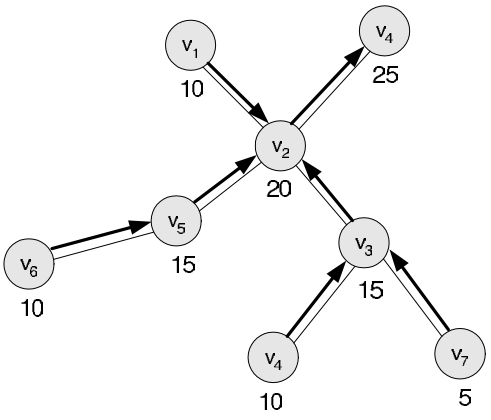


Fig. 6.4 – Construction du graphe de dépendances à partir des mesures de performance

Ainsi, nous illustrons la construction du graphe à l'aide d'un scénario simple à la figure 6.4. Les valeurs d'une métrique de performances sont indiquées en dessous de chaque nœud du réseau. Un arc est établi entre chaque couple de voisins en fonction de ces valeurs : par exemple, le nœud v_5 a une valeur $m(v_5)$ de 15 contre une valeur $m(v_2)$ de 20 pour le nœud v_2, si bien qu'un arc est construit du nœud v_5 vers le nœud v_2. Les données du graphe de dépendances relatives à un nœud donné seront normalisées localement afin qu'il soit possible d'effectuer le calcul distribué de l'importance des nœuds sans renormaliser les valeurs globales à chaque itération.

Calcul centralisé de l'importance d'un nœud

L'objectif consiste ensuite à analyser ce graphe de dépendances pour identifier les nœuds importants en évaluant la centralité des nœuds par vecteur propre [33]. L'algorithme détaillé dans le chapitre précédent ne peut être mis en œuvre tel quel sur des graphes orientés. Dans cette algorithme, l'importance d'un nœud est uniquement calculé en fonction de l'importance des autres nœuds à des coefficients près. Or dans un graphe connecté orienté, un nœud peut ne pas être choisi, comme par exemple le nœud v_7 sur la figure 6.4. La difficulté vient du fait qu'un tel nœud n'a pas d'état (son importance est nulle) et il ne peut contribuer à l'importance des autres nœuds : le nœud v_7 ne peut contribuer à l'importance du nœud v_3 bien qu'un arc les relie. Il est possible de pallier ce problème en modifiant la manière avec laquelle la centralité du nœud est calculée. Jusqu'à présent, le calcul reposait uniquement sur les connexions avec les autres nœuds comme décrit dans l'équation 6.12 (avec $\vec{v}_{pr}^{k+1}$ correspondant à l'approximation du vecteur propre principal $\vec{v}_{pr}$ à la k-ième itération).

$$\vec{v}_{pr}^{k+1} = A^T \times \vec{v}_{pr}^{k}. \tag{6.12}$$

Par ailleurs, nous remarquons que le graphe de dépendances étant orienté, le sens de la matrice a une incidence sur le calcul de la centralité. Ce dernier s'opère à l'aide de la transposée de la matrice d'adjacence A^T puisque l'importance d'un nœud est définie par les arcs entrants de ce nœud. Le problème lié aux nœuds non choisis peut être résolu en considérant que la centralité ne dépend plus uniquement des liens avec les autres nœuds, mais également de l'état interne du nœud à un coefficient près, ce qui aboutit à l'équation 6.13.

$$\vec{v}_{pr}^{k+1} = (1-a) \times A^T \times \vec{v}_{pr}^{k} + a \times \vec{p} \tag{6.13}$$

La centralité est décomposée en un facteur externe $A^T \times \vec{v}_{pr}^{k}$ et un facteur interne $\vec{p}$ qui ont une importance relative fonction du coefficient $a < 1$ (dont la valeur est discutée dans [33]). Ainsi, comme le nœud v_7 n'est pas choisi dans l'exemple précédent, le facteur externe est de valeur nulle. En revanche, le facteur interne lui permet de conserver un état et de contribuer à celui d'autres nœuds, en l'occurence, le nœud v_3. L'algorithme 3 spécifie le nouveau calcul de la centralité par vecteur propre pour l'évaluation de l'importance des nœuds en fonction des métriques de performance. Le vecteur propre principal obtenu à la fin de son exécution nous indique la centralité des nœuds ad-hoc : le i-ème élément du vecteur fournissant la centralité du nœud v_i. Dans cet algorithme, le calcul de la centralité est défini pour être exécuté de manière centralisée par le gestionnaire principal.

Calcul distribué de l'importance d'un nœud

Une version distribuée [117] du calcul de l'importance d'un noeud est décrite dans l'algorithme 4. En fait, chaque nœud v_i peut calculer sa propre valeur de centralité comme suit :

Algorithme 3 : Calcul centralisé de la centralité pour l'identification de nœuds importants

Données : Matrice d'adjacence A
Résultat : Vecteur propre principal $\vec{v}_{pr}$
initialisation
a) initialiser le vecteur solution avec des valeurs identiques pour chaque entrée ;
$\vec{p} = \frac{1}{n} \times \vec{1}$
$\vec{v}^0_{pr} = \vec{p}$
répétition
b) calculer le nouveau vecteur $\vec{v}_{pr}$ en appliquant la matrice d'adjacence A ;
$\vec{v}^{k+1}_{pr} = A^T \times \vec{v}^k_{pr}$
c) prendre en considération l'état interne du nœud à un coefficient a près ;
$\vec{v}^{k+1}_{pr} = (1-a) \times \vec{v}^{k+1}_{pr} + a \times \vec{p}$
d) normaliser le vecteur $\vec{v}_{pr}$ par la somme de ses coefficients ;
e) estimer la convergence en comparant $\vec{v}^k_{pr}$ et $\vec{v}^{k+1}_{pr}$;
$\delta = \| \vec{v}^{k+1}_{pr} - \vec{v}^k_{pr} \|$
jusqu'à
f) la convergence est effective i.e. $\delta < \epsilon$

$$\vec{v_{pr}}_i^{\,k+1} = (1-a)(A_{1i} \times \vec{v_{pr}}_1^{\,k} + ... + A_{ni} \times \vec{v_{pr}}_n^{\,k}) + a \times \vec{p_i} \tag{6.14}$$

Nous constatons qu'il s'agit d'une décomposition élément par élément de l'équation 6.13. Les arcs sont établis entre nœuds voisins durant la construction du graphe de dépendances, ainsi la plupart des coefficients A_{ji} pour un nœud v_i fixé sont nuls permettant de limiter le coût du calcul de $\vec{v_{pr}}_i^{\,k+1}$. L'algorithme 4 subdivise les nœuds voisins en deux sous-ensembles notés E_i et F_i correspondant respectivement aux voisins ascendants (ayant une valeur de métrique supérieure au nœud courant) et aux voisins descendants (ayant une valeur de métrique inférieure au nœud courant).

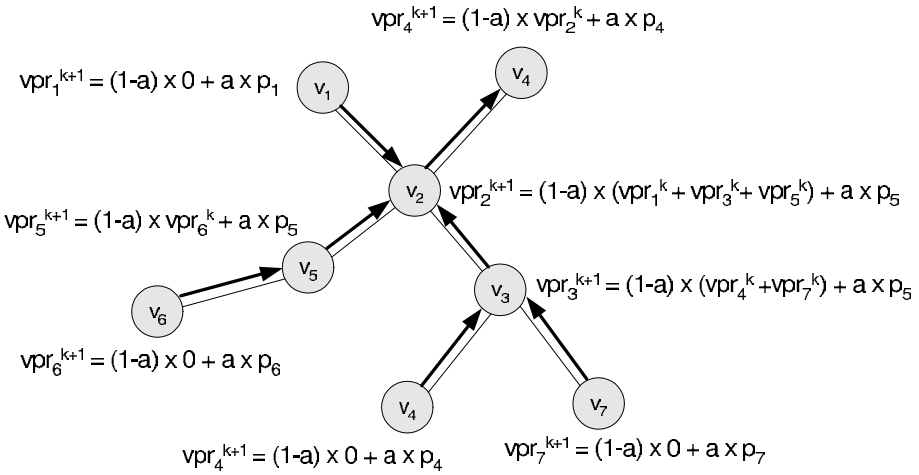


FIG. 6.5 – Application de l'algorithme distribué pour l'identification des nœuds importants

A chaque itération, le nœud v_i calcule sa propre valeur de centralité (i-ème élément du vecteur propre principal) et la transmet aux nœuds ascendants de E_i. Si la valeur n'a pas atteint la convergence, il attend les valeurs de centralité des nœuds descendants de F_i pour recalculer

Algorithme 4 : Calcul distribué de la centralité pour l'identification de nœuds importants

Données : Ensemble des valeurs $m(v_j)$ avec $v_j \in N(v_i)$
Résultat : i-ème élément du vecteur propre principal $\vec{v}_{pr}$
définitions
E_i : ensemble des nœuds voisins ascendants
F_i : ensemble des nœuds voisins descendants
pour chaque nœud v_i, faire {
initialisation
a) interroger les nœuds voisins $v_j \in N(v_i)$ pour connaître les valeurs $m(v_j)$
b) établir les valeurs de A_{ji} et A_{ij} en comparant $m(v_j)$ avec $m(v_i)$
$\forall v_j \in N(v_i),\ si\ m(v_j) > m(v_i)\ alors\ A_{ji} = 1$;
$sinon\ A_{ij} = 1$;
c) construire les ensembles E_i et F_i
$\forall v_j \in N(v_i),\ si\ m(v_j) > m(v_i)\ alors\ E_i = E_i \cup \{v_j\}$;
$sinon\ F_i = F_i \cup \{v_j\}$;
répétition
d) calculer la nouvelle valeur de centralité $\vec{v_{pr}}_i^{\,k+1}$ du nœud v_i ;
$\vec{v_{pr}}_i^{\,k+1} = (1-a)(A_{1i} \times \vec{v_{pr}}_1^{\,k} + ... + A_{ni} \times \vec{v_{pr}}_n^{\,k}) + a \times \vec{p}_i$
e) envoyer la valeur $\vec{v_{pr}}_i^{\,k+1}$ à tous les nœuds dans E_i ;
f) estimer la convergence en comparant $\vec{v_{pr}}_i^{\,k}$ et $\vec{v_{pr}}_i^{\,k+1}$;
$\delta = \| \vec{v_{pr}}_i^{\,k+1} - \vec{v_{pr}}_i^{\,k} \|$
g) attendre que l'ensemble des nœuds de F_i ait retourné leur valeur $\vec{v_{pr}}_j^{\,k+1}$;
jusqu'à
h) la convergence est effective i.e. $\delta < \epsilon$
}

une nouvelle valeur. Ainsi, avec le scénario de la figure 6.5, le nœud v_3 dispose d'un voisin ascendant correspondant au nœud v_2 ($m(v_2) > m(v_3)$) et de deux voisins descendants $\{v_4, v_7\}$ ($m(v_4) < m(v_3)$ et $m(v_7) < m(v_3)$). Le nœud v_3 utilise les valeurs transmises par les nœuds v_4 et v_7 pour calculer sa propre valeur et envoie cette dernière au nœud v_7. L'algorithme distribué permet de réduire la charge de traitement du nœud gestionnaire et de la répartir parmi les nœuds du réseau. La charge de calcul supplémentaire peut néanmoins être refusée par certains nœuds. Nous rappelons que nous proposons dans ce contexte de normaliser les valeurs locales relatives à un nœud (graphe de dépendances) afin de permettre la convergence de cette version distribuée sans renormaliser les valeurs globales à chaque itération.

Une solution intermédiaire consiste à déléguer le calcul de la centralité pour un nœud ad-hoc donné auprès du gestionnaire local, mais à conserver une exécution distribuée du calcul. L'exécution de l'algorithme 4 peut tout à fait être réalisée par un nœud autre que le nœud courant v_i. Mais, ce dernier doit pour ce faire disposer de certains paramètres d'initialisation qui sont propres au nœud v_i. Ainsi lors de la délégation, le nœud courant v_i doit transmettre les ensembles E_i et F_i au nœud gestionnaire local. Celui-ci pourra ensuite contacter les nœuds descendants et ascendants de v_i pour évaluer l'importance du nœud de façon distribuée.

En reprenant le scénario de la figure 6.4, nous considérons le nœud v_2 comme le gestionnaire local et le nœud v_3 comme l'entité dont nous souhaitons déterminer l'importance. Le nœud v_3 envoie les ensembles $E_3 = \{v_4, v_7\}$ et $F_3 = \{v_2\}$ au nœud v_2 qui exécutera l'algorithme 4 à partir de l'étape d) en contactant les nœuds v_4 et v_7. Dans le cas extrême où les étapes d'initialisation

(la construction des ensembles E_i et F_i) ne peuvent pas être prises en charge par le nœud courant v_i, le nœud gestionnaire peut exécuter l'algorithme dans son intégralité en ne disposant uniquement que de la mesure $m(v_i)$ et de la liste des voisins $N(v_i)$.

Nos méthodes d'analyse des mesures de performances reposent d'une part sur une analyse à base de filtres permettant de mettre en évidence les différences et similitudes au sein du réseau ad-hoc, et d'autre part sur une analyse plus fine à base de graphes de dépendances permettant d'identifier dynamiquement les nœuds importants relativement à une métrique de performance.

6.4 Résultats expérimentaux

Nous présentons dans cette section une évaluation de nos méthodes d'analyse à base de filtres et de graphes de dépendances à travers différents scénarios d'expérimentation. Les simulations ont été réalisées à l'aide du simulateur à événement discret ns-2 [74] avec le protocole IEEE 802.11 pour la couche d'accès au médium et le protocole proactif OLSR pour la couche de routage. Nous avons considéré un réseau ad-hoc composé de 100 nœuds (10 x 10 nœuds) disposés sur une surface de 1500 m de côté selon une grille où chaque nœud est situé à une distance de 150 mètres de ses voisins. La mobilité des nœuds est choisie suffisamment faible (par exemple,

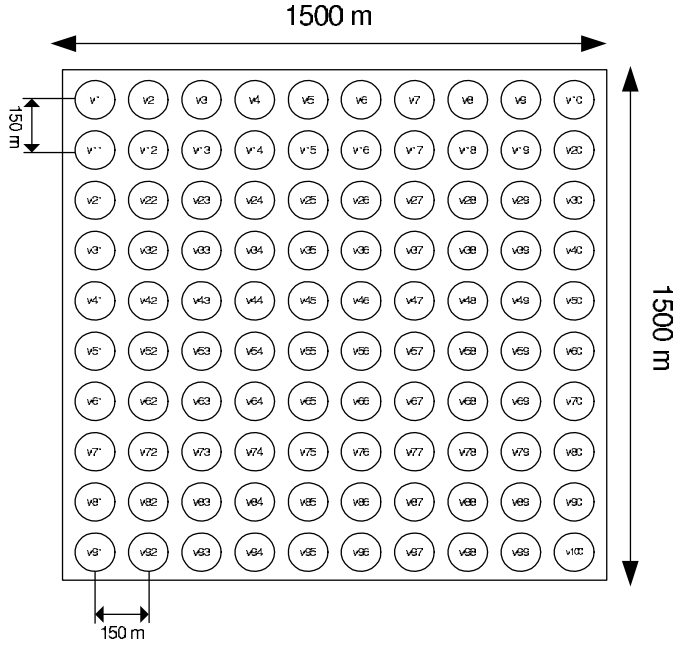


Fig. 6.6 – Topologie réseau considérée durant les expérimentations

dans le contexte de réseaux maillés sans fil), de sorte que la position relative des nœuds reste inchangée durant le temps de simulation de 300 secondes. Les paramètres de simulation communs pour les différents scénarios sont récapitulés dans le tableau 6.4. L'objectif est de déterminer s'il est possible d'identifier sur un réseau ad-hoc suffisamment large des patrons de trafic ou de comportements et d'identifier les nœuds importants du réseau. Plusieurs expérimentations correspondant à des modèles de trafics et de comportements différents vont être décrits à travers cinq scénarios. Pour chaque scénario, nous analyserons les résultats obtenus à travers trois tableaux correspondant respectivement à l'application du filtre 1 de différence, à l'application du filtre 2 de similitude et à l'application de la méthode d'analyse à base de graphes de dépendances.

Paramètres	Valeurs
Simulateur	ns-2
Temps de simulation	300 s
Surface de simulation	1500 m x 1500 m
Nombre de nœuds ad-hoc	100 nœuds
Couche physique	FSP / 2-RGR
Couche MAC	IEEE 802.11
Couche de routage	NRL OLSR
Couche transport	UDP

TAB. 6.1 – Paramètres de simulation

6.4.1 Etablissement d'un *backbone*

Le premier scénario décrit le cas de l'établissement d'un *backbone* [196]. Nous entendons par *backbone* de trafic, un chemin du réseau ad-hoc fortement utilisé où se concentre l'activité de routage. Pour établir ce *backbone*, nous avons considéré un modèle de trafic particulier : les nœuds v_1 et v_{100} sur la topologie précédente s'échangent des données avec un débit moyen de 2 Mbps tandis que les échanges de données des autres nœuds sont définis aléatoirement à l'aide de l'outil *cbrgen* avec un nombre maximum de connections fixé à 100 et un débit moyen fixé à 0.5 Mbps. Durant les expériences, nous avons mesuré pour chaque nœud v_i la métrique de participation au routage $PR(v_i)$ et avons ensuite appliqué nos méthodes d'analyse, telles que présentées à la figure 6.7. Les résultats seront systématiquement organisés de la manière suivante : le schéma en haut à gauche illustre le scénario considéré, le tableau en haut à droite décrit l'analyse à l'aide du filtre 1, le tableau en bas à gauche représente les résultats obtenus avec le filtre 2 et enfin le tableau en bas à droite dépeint les nœuds identifiés comme les plus importants par l'analyse de graphes de dépendances. Les trois principaux nœuds détectés par la méthode sont marqués par ordre d'importance d'une lettre : A, B ou C.

Le filtre 1 de différence fait clairement apparaître un *backbone* sur la diagonale qui s'étend du nœud v_{12} au nœud v_{89} avec une forte activité de routage, ce qui correspond en fait au plus court chemin entre les deux nœuds considérés v_1 et v_{100}. Le filtre 2 de similitude traduit le fait que l'activité de routage ne se limite pas nécessairement à ce sous-ensemble de huit nœuds mais peut également mettre en œuvre des nœuds situés en bordure de l'axe principal tels que les nœuds v_{34} et v_{46} par exemple. Nous noterons en outre que les nœuds à l'origine du trafic sont également mis en évidence par les niveaux de gris. Dans ce scénario, le filtre 2 offre une vue plus détaillée du *backbone* alors que le filtre 1 minimise l'importance des nœuds en bordure du *backbone*. Ceci s'explique par la nature même du filtre : un nœud est d'autant plus important que les nœuds voisins sont faibles. Ainsi, l'activité de routage du nœud v_{34} en bordure est minimisée par la présence du nœud v_{37} localisé directement sur l'axe et ayant une activité plus importante. Le résultat de l'analyse du graphe de dépendances caractérise une activité forte des nœuds de l'axe principal qui sont voisins directs des nœuds v_1 et v_{100}, à savoir les nœuds v_{12} et v_{89}.

6.4.2 Connexion à une passerelle Internet

Le second scénario porte sur la connexion des nœuds ad-hoc à une passerelle Internet [158]. Nous avons désigné arbitrairement un nœud situé au centre de la topologie, en l'occurence le nœud v_{45}, pour intervenir comme une passerelle pour les autres nœuds du réseau. Le modèle de trafic établit des connexions aléatoirement entre les nœuds du réseau et la passerelle, avec

un nombre maximum de connexions fixé à 100 et un débit moyen fixé à 0.5 Mbps. Les résultats obtenus après application des méthodes d'analyse sur la métrique de participation au routage sont décrits à la figure 6.8. Le filtre 2 de similitude met en évidence une activité de routage dans le voisinage direct du nœud passerelle mais le filtre 1 permet de distinguer de manière plus fine le nœud passerelle v_{45} situé à l'intersection des deux diagonales de routage. Ainsi, nous constatons avec le filtre 1 que les nœuds les plus actifs s'étendent sur la première diagonale du nœud v_{12} au nœud v_{89} et sur un deuxième axe très proche de la seconde diagonale du nœud v_{19} au nœud v_{72}. L'analyse du graphe de dépendances identifie le nœud passerelle v_{45} comme le nœud le plus important, suivi respectivement de deux de ces voisins directs : le nœud v_{56} et le nœud v_{36}. Cette fois-ci, nous pouvons considérer que le filtre 1 a offert une meilleure analyse que le filtre 2, avec des résultats fortement corrélés avec l'analyse du graphe de dépendances.

6.4.3 Nœud non collaboratif

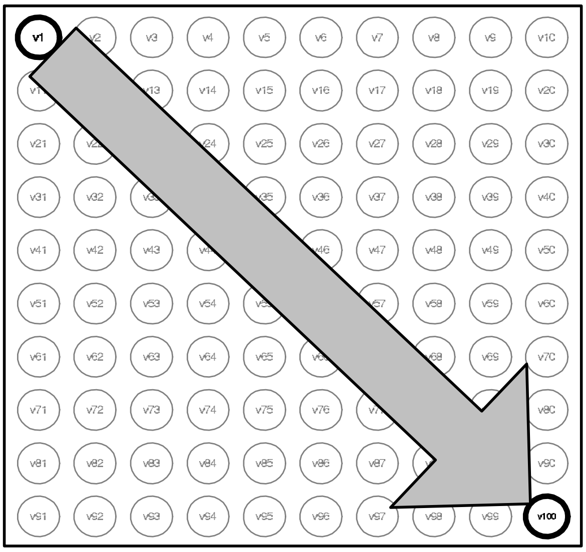
Le troisième scénario prend en considération les même paramètres que le scénario 1 mais avec un nœud non collaboratif qui refuse de participer à l'activité de routage [192]. Les nœuds v_1 et v_{100} s'échangent des données avec un débit moyen de 2 Mbps tandis que les échanges de données des autres nœuds sont définis aléatoirement avec un nombre maximum de connections fixé à 100 et un débit moyen fixé à 0.5 Mbps. Un nœud ad-hoc v_{34} situé sur le *backbone* refuse de router le trafic pour le compte d'autres nœuds. La figure 6.9 présente les résultats obtenus après application des méthodes sur la métrique de participation au routage. Le filtre 1 distingue le nœud v_{34} par une valeur nulle (nœud blanc) dans un voisinage actif (nœuds gris clair et gris foncé). La non collaboration du nœud est compensée par une activité de routage plus importante dans son voisinage, notamment des nœuds v_{25} et v_{43}. Le filtre 2 permet également d'identifier le nœud non collaboratif mais de manière moins évidente puisque la participation des voisins compense partiellement l'inactivité du nœud : la présence de nœuds voisins forts contribue à l'importance du nœud.

6.4.4 Comportement de groupes

Dans le scénario suivant, nous souhaitions étudier des comportements de groupes [139]. Pour ce faire, nous avons limité les échanges de données à deux groupes de 16 nœuds ad-hoc situés dans le coin inférieur gauche et le coin supérieur droit de la topologie, comme décrit sur la figure 6.10. Les échanges définis aléatoirement avec un nombre total de connexions de 100 et un débit moyen total fixé à 0.5 Mbps sont toujours internes à un groupe : un nœud d'un groupe communique uniquement avec un nœud du même groupe. Une exception est faite pour les nœuds v_{82} et v_{19} qui peuvent communiquer ensemble malgré la règle. Les résultats, après analyse de la participation au routage, sont présentés à la figure 6.10. Le filtre 1 caractérise les deux groupes par deux sous-ensembles correspondant au noyau de chacun des groupes : ces nœuds sont au centre du groupe et par conséquent sont plus souvent sollicités pour le routage. Le filtre 2 permet, quant à lui, de colorer l'intégralité des nœuds des groupes et de mettre en évidence la communication entre les nœuds v_{82} et v_{19}. Ainsi, un chemin apparaît entre les deux groupes grâce aux nœuds intermédiaires v_{65}, v_{56} et v_{46}. L'analyse du graphe de dépendances met en évidence les nœuds v_{74} et v_{37} servant de paserelles pour les communications inter-groupes.

6.4.5 Dégradation de l'atteignabilité

Le dernier scénario visait à caractériser une dégradation de l'atteignabilité des nœuds. Nous avons généré un trafic aléatoire entre les nœuds avec un nombre de connexions de 50 et un débit



(a) Scénario 1

0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0
0	0	2	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	2	0	0	0
0	0	1	0	0	0	0	2	0	0
0	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	0	0

(b) Analyse à l'aide du filtre 1

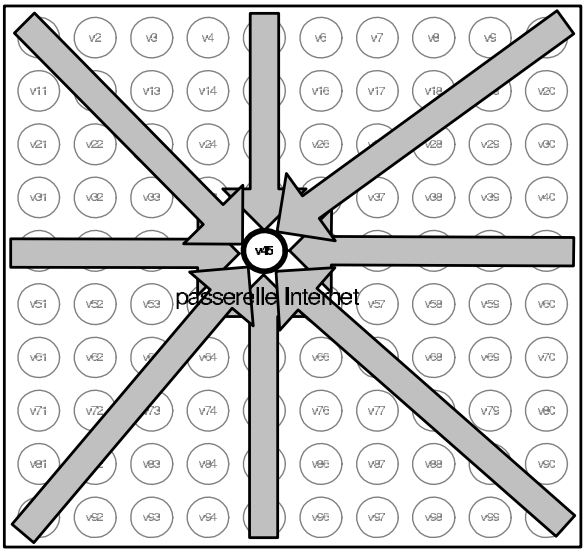
1	0	0	0	0	0	0	0	0	0
0	2	1	0	0	0	0	0	0	0
0	1	2	1	0	0	0	0	0	0
0	0	1	2	1	1	0	0	0	0
0	0	0	1	2	1	1	0	0	0
0	0	0	0	1	2	1	0	0	0
0	0	0	0	0	1	2	1	0	0
0	0	1	0	0	0	1	2	1	0
0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	1

(c) Analyse à l'aide du filtre 2

0	0	0	0	0	0	0	0	0	0
0	A	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	C	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	B	0
0	0	0	0	0	0	0	0	0	0

(d) Analyse du graphe de dépendance

FIG. 6.7 – Résultats obtenus avec le scénario correspondant à l'établissement d'un *backbone* [Les nœuds v_1 et v_{100} s'échangent des données avec un débit moyen supérieur à celui du reste du réseau. Les filtres de différence 1 et de similude 2 mettent clairement en évidence un *backbone* sur la diagonale du nœud v_{12} à v_{89}.]



(a) Scénario 2

0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	1	2	2	2	0
0	0	2	0	0	0	2	0	0	0
0	0	0	2	1	2	2	1	0	0
0	0	0	1	2	0	0	0	0	0
0	0	0	2	1	2	1	0	0	0
0	1	2	0	0	0	2	1	0	0
0	2	0	1	0	0	0	2	1	0
0	0	0	0	0	0	0	2	2	0
0	0	0	0	0	0	0	0	0	0

(b) Analyse à l'aide du filtre 1

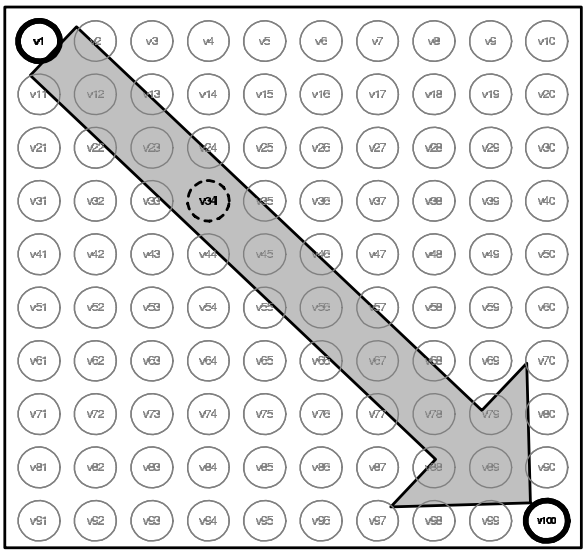
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	0	0
0	0	1	1	1	1	2	1	0	0
0	0	1	2	2	2	2	1	0	0
0	0	0	2	2	2	1	0	0	0
0	0	1	2	2	2	1	1	0	0
0	0	2	1	1	1	2	1	0	0
0	1	1	0	0	1	1	2	1	0
0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0

(c) Analyse à l'aide du filtre 2

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	C	0	0	0	0
0	0	0	0	A	0	0	0	0	0
0	0	0	0	0	B	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(d) Analyse du graphe de dépendance

FIG. 6.8 – Résultats obtenus avec le scénario correspondant à la connexion à une passerelle Internet [Les nœuds se connectent à une passerelle Internet représentée par le nœud v_{45} localisé au centre de la topologie. Le nœud passerelle est facilement identifiable par le filtre 1 à l'intersection de deux axes de routage sur la première et la seconde diagonale.]



(a) Scénario 3

0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0
0	0	2	1	1	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0
0	0	1	1	2	1	0	0	0	0
0	0	1	1	1	2	0	0	0	0
0	0	0	0	0	0	2	0	0	0
0	0	0	0	0	0	1	2	0	0
0	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	0	0

(b) Analyse à l'aide du filtre 1

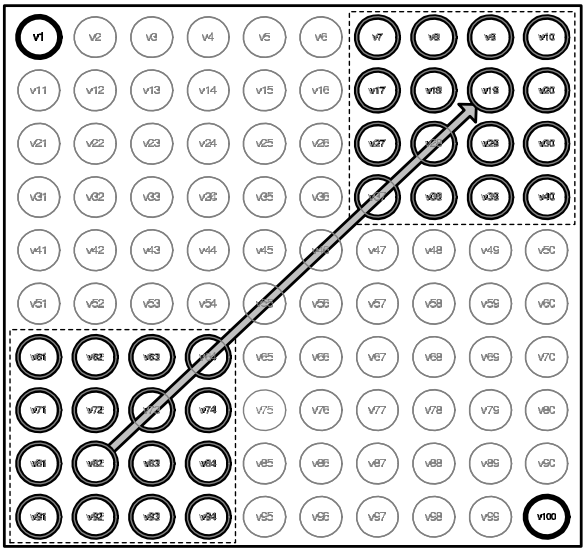
1	0	0	0	0	0	0	0	0	0
0	2	1	0	0	0	0	0	0	0
0	1	2	1	1	0	0	0	0	0
0	0	1	1	2	1	0	0	0	0
0	0	1	2	2	2	1	0	0	0
0	0	1	1	2	2	1	0	0	0
0	0	0	0	1	1	2	1	0	0
0	0	0	0	0	0	1	2	1	0
0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	1

(c) Analyse à l'aide du filtre 2

0	0	0	0	0	0	0	0	0	0
0	A	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	C	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	B	0
0	0	0	0	0	0	0	0	0	0

(d) Analyse du graphe de dépendance

FIG. 6.9 – Résultats obtenus avec le scénario correspondant à un nœud non collaboratif [Les nœuds v_1 et v_{100} s'échangent des données avec un débit moyen supérieur à celui du reste du réseau mais un nœud v_{20} situé sur l'axe de trafic refuse de participer à l'activité de routage. Le filtre 1 identifie le nœud non collaboratif par une valeur nulle (nœud blanc) dans un voisinage actif (nœuds gris clair et gris foncé).]



(a) Scénario 4

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	2	0
0	0	0	0	0	0	0	2	2	0
0	0	0	0	0	0	2	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	2	1	2	0	0	0	0	0	0
0	2	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(b) Analyse à l'aide du filtre 1

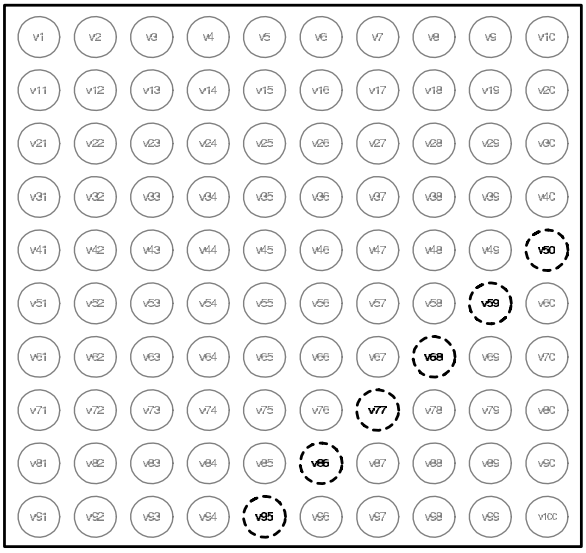
0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	1	2	2	1
0	0	0	0	0	0	1	2	2	1
0	0	0	0	0	0	2	1	1	1
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	2	2	2	0	0	0	0	0	0
1	2	2	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

(c) Analyse à l'aide du filtre 2

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	B	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	C	A	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(d) Analyse du graphe de dépendance

FIG. 6.10 – Résultats obtenus avec le scénario correspondant à un comportement de groupes [Les échanges définis aléatoirement sont toujours internes à un groupe : un nœud d'un groupe communique uniquement avec un nœud du même groupe, à l'exception des nœuds v_{82} et v_{19} qui peuvent communiquer ensemble. Le filtre 2 permet d'identifier les deux nœuds des groupes et de mettre en évidence la communication inter-groupes entre les nœuds v_{82} et v_{19}.]



(a) Scénario 5

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	2	0
0	1	1	1	1	1	1	2	0	0
0	1	1	1	1	1	2	0	0	0
0	1	1	1	1	2	0	0	0	0
0	0	0	0	1	0	0	0	0	0

(b) Analyse à l'aide du filtre 1

2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	1
2	2	2	2	2	2	2	2	1	0
2	2	2	2	2	2	2	1	0	0
2	2	2	2	2	2	1	0	0	0
2	2	2	2	2	1	0	0	0	0

(c) Analyse à l'aide du filtre 2

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	B	0	0
0	0	0	0	0	0	C	0	0	0
0	0	0	0	0	A	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(d) Analyse du graphe de dépendance

FIG. 6.11 – Résultats obtenus avec le scénario correspondant à une dégradation de l'atteignabilité [La participation au routage est volontairement désactivée pour les nœuds situés sur un axe parallèle à la diagonale principale, du nœud v_{95} au nœud v_{50}. Le refus de participer au routage entraîne un isolement entre les deux ensembles de nœuds et provoque un fort degré d'atteignabilité pour les nœuds situés sur l'axe.]

moyen fixé à 0.5 Mbps. Mais nous avons volontairement désactivé l'activité de routage pour les nœuds situés sur un axe parallèle à la seconde diagonale, du nœud v_{95} au nœud v_{50}. Les résultats décrits à la figure 6.11 sont obtenus en appliquant les méthodes d'analyse sur la métrique de degré d'atteignabilité. Nous constatons que le filtre 1 met en évidence un fort degré d'atteignabilité pour les nœuds situés à gauche de la diagonale contre un faible degré pour les nœuds situés à droite de la diagonale. En fait, le refus de participer au routage entraîne un isolement entre les deux ensembles de nœuds : l'ensemble à droite étant composé d'un nombre moins important de nœuds que l'ensemble à gauche, il dispose de valeurs plus faibles. Les nœuds sur l'axe ont les plus fortes valeurs car ils sont en bordure des deux ensembles et peuvent contacter l'ensemble des nœuds du réseau. Evidemment, le résultat observé dépend de la topologie considérée : si la distance entre les nœuds proches de l'axe avait été davantage réduite, comme par exemple la distance entre les nœuds v_{67} et v_{78}, un routage aurait pu être opéré au delà de l'axe. L'analyse du graphe de dépendances identifie les nœuds v_{86}, v_{68} et v_{77} de l'axe comme les nœuds les plus importants en cohérence avec l'analyse du filtre 1.

6.5 Synthèse

Ce chapitre présente une adaptation de la gestion de performances à l'environnement des réseaux ad-hoc. A l'opposé des infrastructures fixes classiques, les éléments d'un réseau ad-hoc n'ont généralement pas de fonctions spécifiques. Leurs propriétés ainsi que leurs tâches évoluent dynamiquement dans le temps : une entité mobile pouvant aléatoirement intervenir comme terminal et comme routeur pour le compte d'autres entités. De nouvelles méthodes d'analyse sont requises pour offrir une vue synthétique de l'état fonctionnel du réseau.

Nous avons tout d'abord introduit deux métriques de performances permettant de tenir compte de la bivalence de l'activité d'un nœud ad-hoc : le degré d'atteignabilité qui permet d'évaluer la capacité d'un nœud à communiquer avec les autres nœuds du réseau, et la participation au routage qui permet de caractériser les nœuds participant activement au fonctionnement du réseau.

Nous avons ensuite défini deux méthodes d'analyse permettant d'extraire de la connaissance sur le réseau à partir des mesures de performance. La première méthode consiste à traiter les mesures à l'aide d'un filtrage de similitude ou de différence pour faire apparaître des patrons de trafics et de comportements, notamment identifier des *backbones* où l'activité de routage se concentre. La seconde méthode offre une analyse plus fine à base de graphes de dépendances permettant d'automatiser la détection des nœuds importants au sein de la structure du réseau. Les méthodes d'analyse à base de filtres et à base de graphes ont été spécifiées mathématiquement sous la forme d'algorithmes.

Ces méthodes ont également été expérimentées à travers un ensemble de cinq scénarios différents : établissement d'un *backbone*, connexion à une passerelle Internet, nœud non collaboratif, comportement de groupes et dégradation de l'atteignabilité dans un réseau. Elles ont permis de mettre en évidence l'intérêt de notre approche de gestion de performances pour mesurer et synthétiser l'état fonctionnel d'un réseau ad-hoc. Si ces méthodes permettent d'évaluer l'impact des nœuds au sein du réseau, des méthodes spécifiques sont également requises pour identifier les pathologies au sein du réseau dans le cadre de la gestion de fautes.

Les travaux présentés dans ce chapitre ont fait l'objet de plusieurs publications [12, 20, 14, 21].

Chapitre 7

Gestion de fautes par inférence

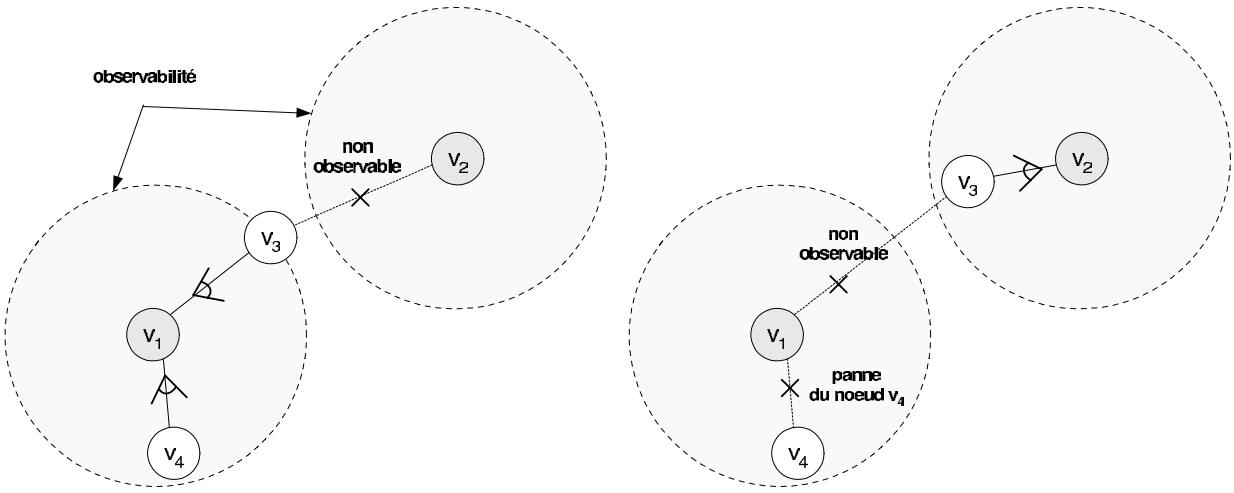
Sommaire

7.1 Introduction

La seconde aire fonctionnelle que nous avons étudiée afin d'en adapter les opérations de gestion correspond à la gestion de fautes [178]. Si la gestion de fautes est un problème bien connu dans les réseaux fixes classiques, la nature fortement dynamique des réseaux ad-hoc amène à repenser cette activité. La gestion de fautes a pour but la détection, l'isolation et la correction éventuelle des anomalies qui affectent le fonctionnement des réseaux et de leurs services. Contrairement aux réseaux fixes, la détection de fautes peut être contrainte dans un réseau ad-hoc par l'impossibilité d'observer un nœud à un instant donné.

L'observabilité est un problème majeur avec les réseaux ad-hoc. Dans un réseau fixe, un équipement peut raisonnablement être considéré comme non fonctionnel lorsqu'il ne répond pas à un appel. En revanche, un nœud ad-hoc peut ne pas répondre parce qu'il n'a temporairement plus de connectivité suite à un déplacement. Un exemple est présenté sur la figure 7.1, où un nœud v_1 observe un nœud v_3 et un nœud v_4. L'observabilité (représentée par le symbole $\lhd$) est considérée dans notre cas comme une relation de voisinage direct. Bien que cette notion d'observabilité peut

sembler limitée et s'inscrire dans un contexte de réseaux agnostiques (ignorant la connectivité multi-sauts), nous considérons que cette définition est préférable dans le cadre d'une détection de fautes fondée sur un monitorage passif. Dans cet exemple, nous nous plaçons du point de vue du nœud v_1. Dans une première phase, les deux nœuds v_3 et v_4 sont perçus comme opérationnels. Dans une seconde phase, le nœud v_3 n'est plus directement connecté à v_1 et le nœud v_4 tombe en panne. Du point de vue du nœud v_1, ces deux situations sont les mêmes. Sans une mesure active additionnelle (ping par exemple), v_1 ne peut distinguer les deux situations.



(a) Réseau ad-hoc à la phase 1 (b) Réseau ad-hoc à la phase 2

FIG. 7.1 – Scénario illustrant le problème d'observabilité lié à la gestion de fautes

Nous proposons dans ce chapitre de prendre en considération le problème d'observabilité et de définir une nouvelle méthode pour la gestion de fautes dans les réseaux ad-hoc. Cette méthode consiste à analyser l'intermittence [172] des nœuds ad-hoc et à détecter des fautes/pannes par inférence. L'intermittence d'un nœud peut être provoquée par des causes bénignes telles que la mobilité ou la dégradation temporaire de la connectivité. Elle devient cependant pathologique lorsqu'elle est causée par des pannes réseau ou par des activités malicieuses. Pannes et comportements intermittents peuvent donc être fortement liés dans les réseaux ad-hoc. Un nœud peut être caractérisé par une intermittence régulière/normale liée à la nature dynamique et à la forte mobilité du réseau. En revanche, une panne peut générer un comportement intermittent que l'on qualifiera de *pathologique*. Un problème clé réside dans le fait de pouvoir différencier une intermittence pathologique d'une intermittence normale, et ainsi de pouvoir distinguer les nœuds ad-hoc en panne de ceux qui sont opérationnels.

Ce travail de recherche s'inscrit dans le cadre d'un monitorage léger et passif des réseaux ad-hoc qui permet d'évaluer l'intermittence des nœuds et de détecter par inférence des fautes et pannes. La consommation de ressources liée à la gestion peut être négligée avec un réseau fixe dans une certaine proportion. En revanche, elle devient d'une importance majeure lorsque l'on se place dans un contexte où énergie et bande passante sont fortement limitées. Notre méthode repose sur un monitorage passif car elle doit générer une charge réseau additionnelle qui soit minimale. Nous observons les informations qui circulent dans le plan de routage et qui seraient nécessairement traitées par les nœuds ad-hoc. Puis, nous en dérivons une mesure fondée sur la théorie de l'information [62] qui permet de détecter l'intermittence pathologique d'un nœud.

Par ailleurs, la fiabilité est un autre problème majeur puisque les nœuds peuvent fournir des

informations erronées ou peuvent volontairement les corrompre à leur avantage. Nous définissons différentes méthodes de détection distribuées où la détection est opérée grâce à la collaboration de plusieurs nœuds ad-hoc. L'objectif est d'offrir une infrastructure de gestion plus robuste qui prend en charge les problèmes liés à une vue locale biaisée - une vue trop spécifique à un nœud donné - aussi bien que les problèmes liés à la propagation d'informations erronées fournies volontairement ou non par des nœuds du réseau.

La suite de ce chapitre est organisée de la manière suivante : nous présentons tout d'abord dans la section 2 le concept d'intermittence et les différentes causes sous-jacentes dans les réseaux ad-hoc. Nous définissons ensuite dans la section 3 une approche légère de monitorage permettant d'identifier l'intermittence pathologique de nœuds ad-hoc et de détecter par inférence des fautes et pannes. Nous y présenterons comment analyser le plan de routage et comment modéliser la perception d'un nœud ad-hoc. Dans ce cadre, nous introduirons une mesure fondée sur la théorie de l'information permettant de caractériser l'intermittence d'un nœud. Différentes méthodes de détection distribuées et collaboratives, incluant une méthode auto-configurable, seront décrites dans la section 4. Enfin, nous terminerons en présentant dans la section 5 un ensemble de résultats de simulation qui nous permettra d'évaluer les performances de notre gestion de fautes.

7.2 Intermittence des nœuds ad-hoc

L'intermittence [172] est un phénomène relativement courant dans les réseaux ad-hoc mais peut être provoquée par deux catégories différentes de causes. Nous distinguons l'intermittence régulière induite par la dynamique même du réseau de l'intermittence pathologique induite par des fautes/pannes en son sein.

La première catégorie correspond à l'intermittence normale qui apparaît à cause de la dynamique du réseau, notamment à cause de la mobilité des nœuds ad-hoc. Nous obtenons ce phénomène lorsqu'un nœud ad-hoc observe son voisinage direct alors que les nœuds se déplacent. Ainsi, deux nœuds peuvent être directement connectés à un instant donné, puis peuvent perdre cette connectivité directe à cause de la mobilité. Cette connectivité peut ensuite être rétablie un peu plus tard lorsqu'ils se trouvent à nouveau dans le même voisinage.

La seconde catégorie est l'intermittence pathologique qui traduit un symptôme pouvant être lié à différentes formes de pathologies. Nous pouvons notamment considérer parmi ces pathologies :

- les fautes dues à une mauvaise configuration et aux erreurs de la couche physique : elles peuvent générer un comportement atypique où les nœuds apparaissent intermittents bien que d'un point de vue géographique ils ne se déplacent pas significativement,
- les pannes de routage : elles sont rencontrées lorsque le processus de routage est perturbé volontairement par une activité malicieuse (attaques du plan de routage) ou involontairement à cause d'erreurs de configuration de la pile protocolaire,
- les cas de mobilité pathologique : bien qu'il soit difficile de définir en soit une mobilité normale, des motifs de mobilité non prédits peuvent avoir un sérieux impact sur le fonctionnement et le niveau de service d'un réseau ad-hoc, en particulier dans des cas d'utilisation spécifiques tels que les applications de défense et de sécurité.

Le problème de notre approche peut se ramener à deux points principaux : peut-on détecter un comportement intermittent pathologique en monitorant des paramètres élémentaires comme par exemple les informations de routage ? Et peut-on effectuer ce monitorage de manière distribuée afin de le rendre plus robuste ?

7.3 Monitorage de l'intermittence des nœuds

Nous proposons de caractériser l'intermittence des nœuds ad-hoc en définissant une mesure fondée sur la théorie de l'information. Cette mesure repose sur une analyse du plan de routage, plus précisément sur une analyse de la distribution des paquets de contrôle. Nous spécifierons un cadre formel fondé sur une modélisation markovienne [37] de la perception d'un nœud, qui servira de base pour la définition et l'étude des propriétés de la mesure.

7.3.1 Analyse du plan de routage

Le monitorage de l'intermittence des nœuds est réalisé de manière passive en analysant le plan de routage. Nous considérons à nouveau comme support le protocole standard pro-actif OLSR [57]. Nous rappelons que chaque nœud OLSR effectue typiquement deux opérations principales : (1) il détermine la liste des nœuds voisins auxquels il est directement lié (un saut) par émission périodique de messages de beaconing HELLO et (2) il échange les informations d'état de liens avec les autres nœuds du réseau par diffusion de messages de contrôle de topologie. OLSR réduit la charge de contrôle en sélectionnant un sous-ensemble de nœuds appelé relais multipoints (MPRs) pour la propagation des messages de contrôle de topologie. Cette sélection des MPRs repose sur une heuristique qui consiste à sélectionner pour un nœud donné le sous-ensemble minimal de voisins à un saut qui permet d'atteindre l'ensemble des nœuds à deux sauts, afin de limiter le nombre de liens utilisés lors de la diffusion des informations de topologie.

Distribution de paquets

Notre gestion de fautes consiste à analyser le protocole de routage OLSR en étudiant la distribution des paquets reçus par chaque nœud durant l'opération (1) de beaconing. Ceci est réalisé afin de détecter les nœuds ayant un comportement intermittent pathologique. En cohérence avec nos précédents travaux de recherche, nous considérons un réseau mobile ad-hoc comme un ensemble de n nœuds mobiles $V = \{v_1, v_2, ..., v_n\}$ se déplaçant sur une surface donnée durant une période de temps T. Cette période T est découpée en k intervalles de mesure $[t_l, t_{l+1}]$ avec $t_l = l \times \frac{T}{k}$ pour un entier $l \in [0, k]$. La durée $[t_l, t_{l+1}]$ doit être plus longue que l'intervalle de temps r entre deux émissions de paquets HELLO par un nœud du réseau, de sorte que plusieurs paquets HELLO peuvent être émis durant l'intervalle $[t_l, t_{l+1}]$.

Le choix d'une valeur optimale pour la période de temps T et pour les intervalles $[t_l, t_{l+1}]$ constitue un travail de recherche à part entière, mais une excellente analyse des échelles de temps pour le flux d'informations de monitorage dans les réseaux ad-hoc peut être trouvée dans [71]. Durant l'opération de beaconing du protocole OLSR, chaque nœud $v_i \in V$ peut recevoir des paquets HELLO des autres nœuds situés dans le voisinage à un saut.

Définition 3 *Le nombre de paquets HELLO de beaconing reçu par un nœud v_i en provenance d'un nœud v_j est noté X_{v_i,v_j} et correspond à une variable aléatoire [109] $X_{v_i,v_j}(l) : [0, k] \rightarrow [0, b_{max}]$ avec l caractérisant l'intervalle de temps $[t_l, t_{l+1}]$ et b_{max} le nombre maximal de paquets HELLO que v_i peut recevoir de v_j durant l'intervalle.*

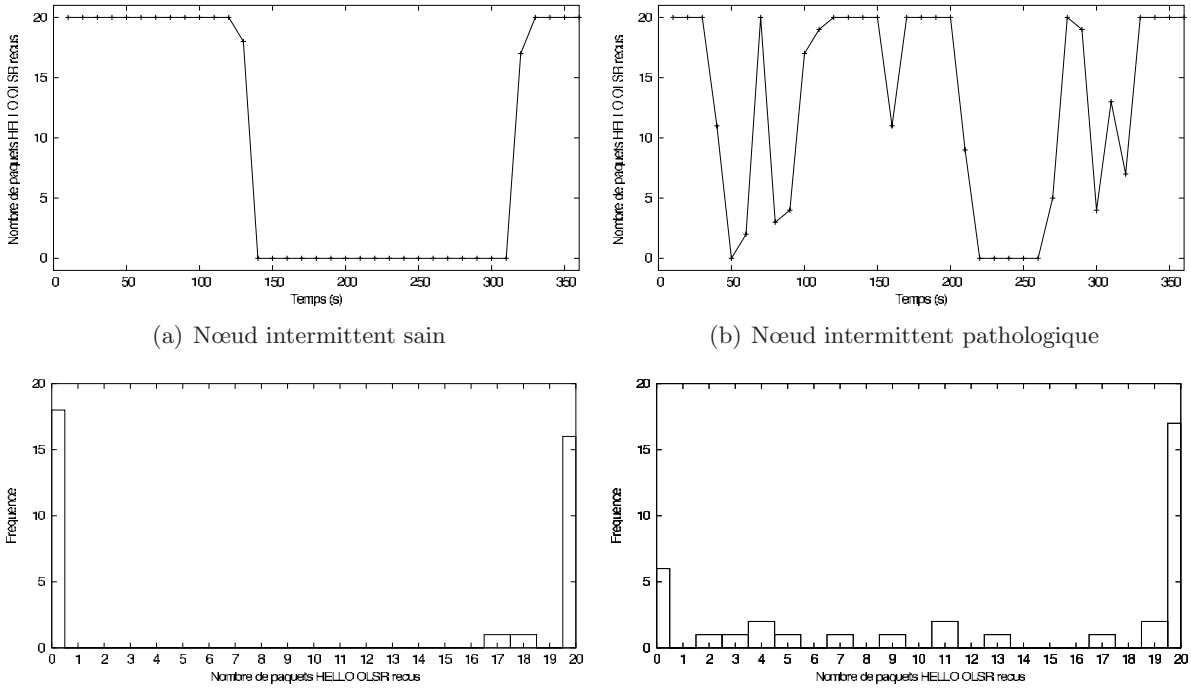
Si l'intervalle de temps constant[3] r entre deux émissions de paquets est supposé homogène entre les nœuds du réseau ad-hoc, alors la variable aléatoire X_{v_i,v_j} prend des valeurs comprises entre 0 et au maximum $b_{max} = \frac{1}{r} \times \frac{T}{k}$. L'homogénéité des valeurs n'est cependant pas une

[3] constante notée HELLO_INTERVAL dans OLSR RFC 3626

contrainte forte, puisque notre approche de monitorage peut facilement être étendue pour des valeurs r hétérogènes entre les nœuds.

Exemples illustratifs

Le monitorage étant réalisé sur le plan de routage, nous pouvons détecter une pathologie provenant de multiples facteurs tels que pannes de routage, problèmes de batterie, perturbations physiques fortes et mobilité pathologique. Le monitorage repose sur l'analyse de la manière avec laquelle un nœud intermittent pathologique est perçu par un nœud voisin ou par un ensemble de nœuds voisins. Une idée intuitive de cette perception peut être donnée en analysant les valeurs $X(v_i, v_j)$ d'un nœud v_i pour différents nœuds v_j du réseau.



(a) Nœud intermittent sain

(b) Nœud intermittent pathologique

(c) Distribution de paquets pour le nœud sain

(d) Distribution de paquets pour le nœud pathologique

FIG. 7.2 – Exemples illustrant le nombre $X(v_i, v_j)$ de paquets HELLO périodiquement reçus par un nœud ad-hoc (sous-figures a et b) avec la distribution de paquets correspondante (sous-figures c et d) [Du point de vue du nœud observateur, un nœud intermittent sain génère une distribution étroite avec la majorité des valeurs égales à 0 et b_{max} (sous-figure c), tandis qu'un nœud intermittent pathologique se distingue par une distribution plus large (sous-figure d).]

Les figures 7.2(a) et 7.2(b) décrivent ces valeurs respectivement pour un nœud intermittent sain et pour un nœud intermittent pathologique. Les graphes ont été obtenus par simulation sur une même période de temps. Notre objectif est ici de fournir un exemple illustratif. Des résultats de simulation plus élaborés seront présentés dans une section suivante. Chaque figure représente le nombre de paquets HELLO reçus $X(v_i, v_j)$ (sur l'axe des ordonnées) durant chaque intervalle de temps $[t_l, t_{l+1}]$ (sur l'axe des abscisses). La courbe ne représente pas la couche physique mais bien une fonction discrète définie à partir de la couche de routage.

A la figure 7.2(a), le nœud ad-hoc sain correspond à une distribution étroite avec la majorité des valeurs égale à 0 lorsque le nœud n'est pas dans le voisinage, et égale à b_{max} lorsque le

nœud se trouve dans le voisinage du nœud observateur. A la figure 7.2(b), le nœud intermittent pathologique (tel que perçu par les autres nœuds) est caractérisé par une distribution plus large des valeurs de $X(v_i, v_j)$. Pour cette illustration, nous avons choisi un comportement fortement pathologique pour faciliter la distinction entre les deux distributions de paquets HELLO. La comparaison directe entre les figures est souvent moins évidente à l'oeil nu dans des scénarios réels. Elle nécessite une analyse formelle, comme nous allons le présenter dans les sections suivantes.

7.3.2 Modélisation markovienne de la perception d'un nœud

La perception d'un nœud intermittent pathologique par un nœud observateur peut être modélisé par une chaîne de Markov discrète [37] à quatre états notés $\{S_1, S_2, S_3, S_4\}$, comme décrit à la figure 7.3.2.

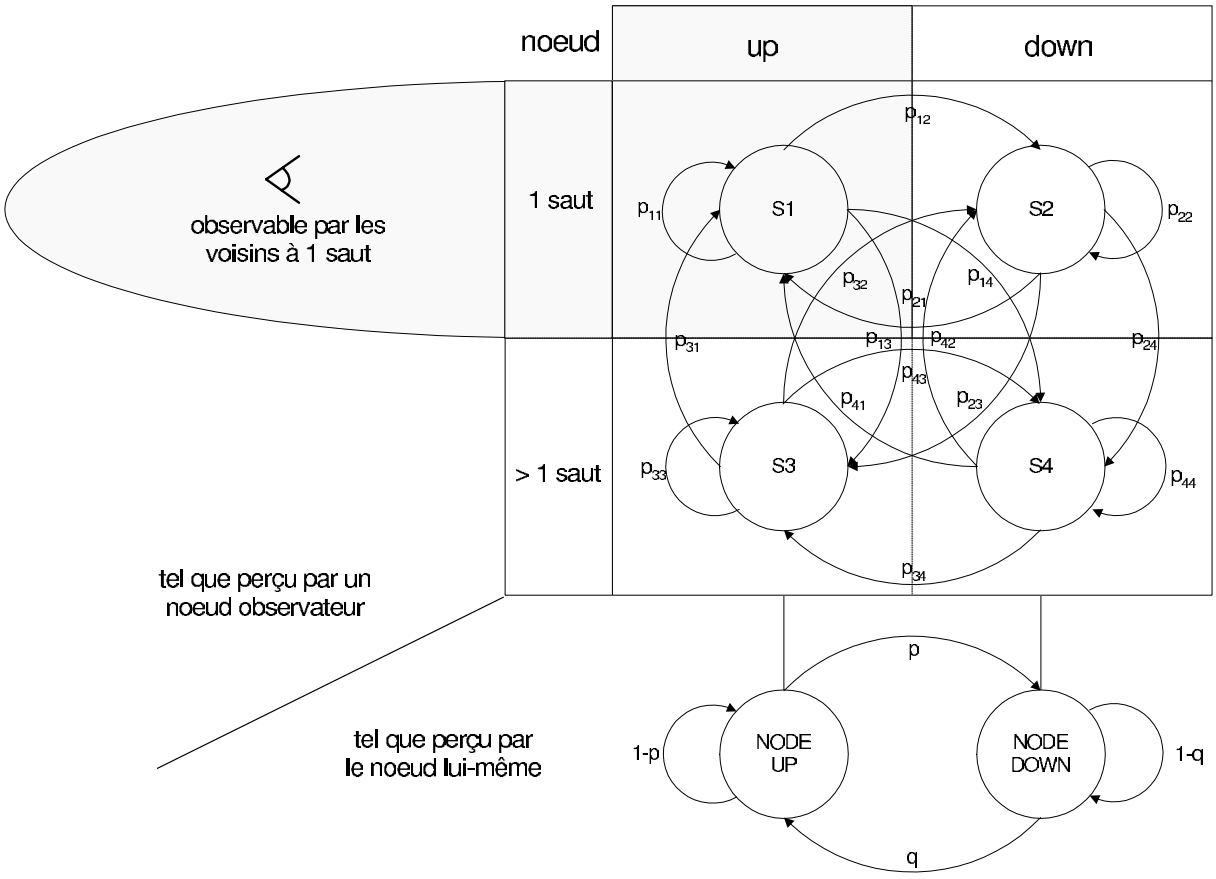


FIG. 7.3 – Modélisation par chaînes de Markov du comportement intermittent pathologique d'un nœud ad-hoc et de sa perception par un nœud observateur

Ces quatre états dépendent de l'état fonctionnel du nœud observé (en fonctionnement ou en panne), mais aussi de sa position par rapport au nœud observateur (voisin à un saut ou non). Un nœud observateur ne peut percevoir un nœud intermittent que si ce dernier est à la fois en fonctionnement et est un voisin à un saut. Cela signifie que le nœud observé doit être dans l'état S_1 de la chaîne de Markov. Supposons les probabilités de transitions p_{ij} d'aller de l'état S_i à l'état S_j et les probabilités d'état $P(S_i)$ d'être dans l'état S_i. La chaîne de Markov sera dans

l'état S_1 soit si la chaîne est déjà dans l'état S_1 et y reste, soit si la chaîne se trouve dans un autre état et transite dans l'état S_1.

Du point du nœud observé, le comportement intermittent pathologique peut se réduire à une chaîne de Markov discrète à deux états, en fonctionnement ou en panne {NODE UP, NODE DOWN} (présentée dans la partie basse de la figure 7.3.2) avec les probabilités de transition p que le nœud tombe en panne et q que le nœud soit réparé après une panne. Les probabilités de transition peuvent ensuite être calculées en utilisant les probabilités conditionnelles, telles que données par les équations 7.1 et 7.2.

$$p = \frac{p(S_1).p_{12} + p(S_1).p_{14} + p(S_3).p_{32} + p(S_3).p_{34}}{P(S_1) + P(S_3)} \tag{7.1}$$

$$q = \frac{p(S_2).p_{21} + p(S_2).p_{23} + p(S_4).p_{41} + p(S_4).p_{43}}{P(S_2) + P(S_4)} \tag{7.2}$$

Cette chaîne de Markov à deux états est irréductible (chaque état est accessible à partir de chacun des autres états) et récurrente positivement (le temps de retour est supposé fini pour chaque état de la chaîne) avec les probabilités $p \neq 0$ et $q \neq 0$. Dans ce cas, il existe une distribution stationnaire unique et, par définition, le processus construit en prenant cette distribution stationnaire comme distribution initiale est ergodique. L'équation de stationnarité peut être résolue pour obtenir la distribution stationnaire unique de cette chaîne de Markov, comme présentée à l'équation 7.3 où p_{up} est la probabilité d'être dans l'état NODE UP et p_{down} est respectivement la probabilité d'être dans l'état NODE DOWN.

$$(p_{up}, p_{down}) = (\frac{q}{p+q}, \frac{p}{p+q}) \tag{7.3}$$

Afin d'évaluer l'impact d'un nœud intermittent pathologique (paramètres p et q) sur la distribution de $X(v_i, v_j)$, nous considérons un simple scénario où v_i et v_j sont dans le même voisinage avec v_i le nœud observateur et v_j le nœud observé qui est intermittent pathologique. Pendant

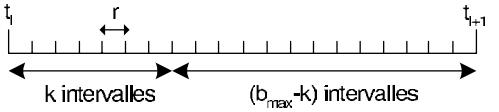


FIG. 7.4 – Intervalle de mesure $[t_l, t_{l+1}]$ divisé en b_{max} intervalles d'émission de paquets HELLO

l'intervalle de mesure $[t_l, t_{l+1}]$, la probabilité pour le nœud v_j d'émettre un paquet HELLO à chaque intervalle d'émission r est donnée par p_{up}, la probabilité que le nœud OLSR v_j soit en fonctionnement. Ainsi, la probabilité pour v_i de recevoir k paquets HELLO (et donc la probabilité de ne pas recevoir $(b_{max} - k)$ paquets HELLO) durant $[t_l, t_{l+1}]$ est caractérisée par une distribution binômiale, définie par l'équation 7.4.

$$P(X_{v_i,v_j} = k) = \binom{k}{b_{max}} p_{up}^k (1 - p_{up})^{b_{max}-k} \tag{7.4}$$

La distribution de probabilité peut être utilisée pour évaluer l'impact des probabilités de transition p et q sur le comportement d'intermittence observé.

7.3.3 Définition et analyse formelles de la mesure

Nous proposons de monitorer le protocole de routage OLSR en réalisant une mesure d'entropie de la distribution de probabilité de $X(v_i, v_j)$. L'entropie, définie par Shannon dans [175], fournit une mesure du désordre d'un système, avec des valeurs élevées caractérisant les systèmes les plus désordonnés. Dans notre cas, elle permet de caractériser le désordre de la distribution (distribution large) des paquets HELLO pour chaque nœud voisin. L'équation 7.5 définit la mesure d'entropie notée $H(X(v_i, v_j))$ de manière formelle.

$$H(X_{v_i,v_j}) = \sum_{k=0}^{b_{max}} P(X_{v_i,v_j} = k).log(\frac{1}{P(X_{v_i,v_j} = k)}) \tag{7.5}$$

Considérons la mesure d'entropie pour les exemples présentés dans les figures 7.2(a) et 7.2(b). $H(X_{v_i,v_j})$ vaut 1.307 pour un nœud intermittent sain, tandis que $H(X_{v_i,v_j})$ atteint 2.642 pour un nœud intermittent pathologique. Des valeurs élevées de $H(X_{v_i,v_j})$ identifient une distribution désordonnée avec des valeurs $X(v_i, v_j)$ couvrant largement l'intervalle de valeurs $[0, b_{max}]$, et donc identifient des nœuds dont l'intermittence est anormale.

En reprenant la distribution discrète $X(v_i, v_j)$ donnée par l'équation 7.4, l'entropie $H(X_{v_i,v_j})$ de cette distribution binômiale peut être approximée asymptotiquement par dépoissonisation analytique, tel que proposé par Jacquet et Szpankowski dans [113] (voir l'équation 7.6 où les valeurs a_k sont des constantes calculables explicitement).

$$\begin{aligned} H(X_{v_i,v_j}) &\asymp \frac{1}{2}ln(b_{max}) + ln\sqrt{2\pi p_{up}(1-p_{up})} \\ &\quad + \sum_{k\geq 1} a_k b_{max}^{-k} \end{aligned} \tag{7.6}$$

$$\asymp ln\sqrt{\frac{2\pi pq}{(p+q)^2}} + c \tag{7.7}$$

Dans l'équation 7.7, la valeur approximée de l'entropie $H(X_{v_i,v_j})$ peut alors être exprimée en fonction des probabilités de transition (p, q) de la chaîne de Markov (en utilisant l'équation 7.3) avec la constante $c = \frac{1}{2}ln(b_{max}) + \sum_{k\geq 1} a_k b_{max}^{-k}$. L'impact des paramètres $(p, q) \in]0, 1[^2$ est estimé en étudiant les dérivées partielles de $H(X_{v_i,v_j})$. Ces dérivées sont présentées dans les équations 7.8 et 7.9, et présentent un comportement symétrique.

$$\frac{\partial H(X_{v_i,v_j})}{\partial p} = \frac{1}{2p} - \frac{1}{p+q} \tag{7.8}$$

$$\frac{\partial H(X_{v_i,v_j})}{\partial q} = \frac{1}{2q} - \frac{1}{p+q} \tag{7.9}$$

L'impact de la probabilité p (que le nœud tombe en panne) sur $H(X_{v_i,v_j})$ pour une probabilité donnée q (que le nœud soit rétabli) est évalué par le tableau de variations de $\partial H(X_{v_i,v_j}) \setminus \partial p$ sur $]0, 1[$. Le tableau montre que $\partial H(X_{v_i,v_j}) \setminus \partial p$ est positive sur l'intervalle $]0, q[$, atteint zéro pour $p = q$ et enfin devient négative sur $]q, 1[$. Ainsi, l'entropie $H(X_{v_i,v_j})$ atteint une valeur maximale $(ln\sqrt{\pi/2} + c)$ pour $p = q$. De manière symétrique, pour une probabilité de transition donnée p, l'entropie $H(X_{v_i,v_j})$ atteint une valeur maximale lorsque la probabilité q de rétablir le nœud vaut p. Ce constat est confirmé graphiquement par la figure 7.5, où l'entropie théorique est tracée comme une fonction $H(X_{v_i,v_j})$ pour des valeurs données de p, en variant la probabilité

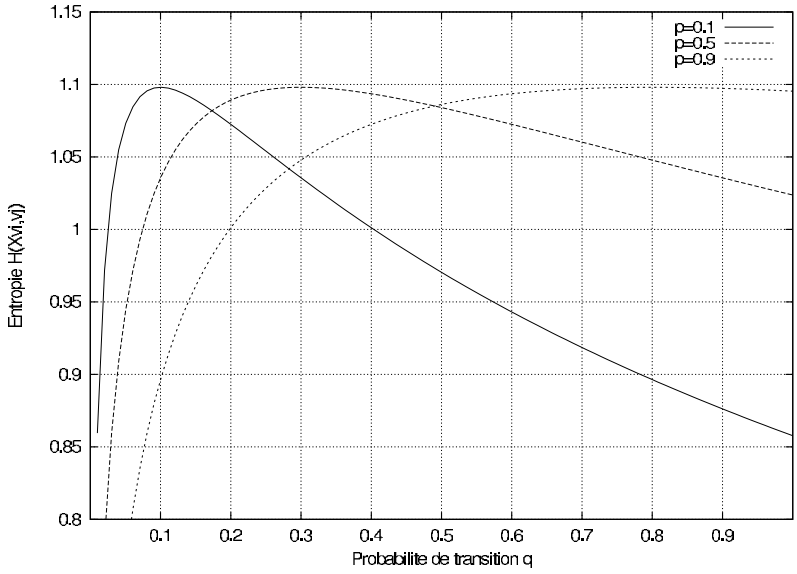


FIG. 7.5 – Courbes représentant l'entropie théorique additionnelle $H(X_{v_i,v_j})$ générée par un nœud intermittent pathologique pour une probabilité de panne donnée p, en variant la probabilité de réparation q

de réparation q de 0.1 à 0.9. Chaque courbe (valeurs constantes $p = 0.1, p = 0.5, p = 0.9$) atteint une valeur maximale lorsque les probabilités p et q sont égales. Dans un scénario réaliste, la probabilité de panne p se résume à une valeur faible (telle que $p = 0.1$ ou $p = 0.2$), tandis que la probabilité de réparation q peut atteindre des valeurs plus élevées ($q > p$). Lorsque l'on observe la fonction entropie dans le cas ($p = 0.1$ et $q > p$), nous constatons que celle-ci décroît lorsque q parcourt l'intervalle $[0.1, 0.9]$ (dérivée partielle $\partial H(X_{v_i,v_j}) \setminus \partial p$ négative pour $q > p$).

Ainsi, l'intermittence pathologique d'un nœud ad-hoc est d'autant plus caractérisée par la mesure d'entropie que la probabilité de réparation est faible. L'approximation probabiliste de l'entropie fournit une quantification de l'entropie additionnelle générée par un nœud intermittent pathologique (additionnelle à celle causée par la mobilité du nœud), perçue localement par un nœud ad-hoc. Cette approximation montre comment les nœuds intermittents pathologiques peuvent être identifiés par un nœud local, en sélectionnant les nœuds du réseau présentant les valeurs les plus élevées d'entropie $H(X_{v_i,v_j})$ pour leur distribution de paquets HELLO. La fiabilité de cette mesure locale peut être améliorée par la collaboration entre plusieurs nœuds ad-hoc.

7.4 Détection de fautes distribuée et collaborative

Nous avons présenté dans la section précédente une mesure entropique de la distribution de paquets HELLO qui permet de détecter localement les nœuds ad-hoc pathologiques. Nous complétons notre travail de recherche en proposant différentes méthodes de détection distribuées et collaboratives qui permettent d'améliorer les performances de la solution en partageant les mesures locales entre les différents nœuds du réseau. Nous détaillerons également un mécanisme d'auto-configuration de la détection fondé sur la méthode des *k-means*.

7.4.1 Principe de fonctionnement

La détection peut être améliorée en partageant les mesures locales entre les nœuds du réseau de manière distribuée. Comme décrit à la figure 7.6, les nœuds ad-hoc $\{v_1, v_2, v_4, v_5, v_6\}$ monitorent localement les nœuds voisins et échangent leurs mesures locales entre eux pour détecter le nœud pathologique v_8. Nous allons présenté plusieurs méthodes distribuées pour synthétiser les mesures locales et fournir un monitorage plus efficace et fiable de l'intermittence à l'échelle du réseau. Pour effectuer les mesures locales de l'intermittence, chaque nœud ad-hoc $v_i \in V$

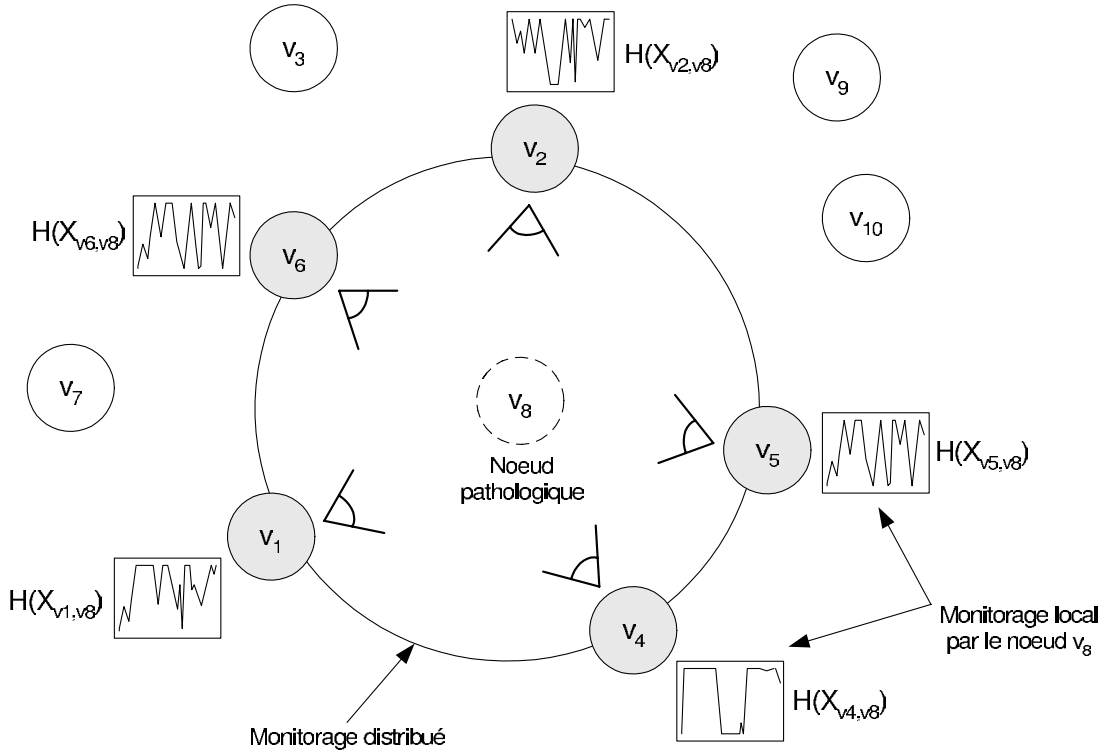


FIG. 7.6 – Monitorage distribué de l'intermittence des nœuds ad-hoc

maintient localement une liste $I(v_i)$ des nœuds qui sont potentiellement pathologiques, avec les mesures entropiques associées à chacun d'eux. L'expression *potentiellement* est employée ici, puisque nous considérons que la mesure locale n'est pas suffisante et doit être confirmée par les mesures des autres nœuds du réseau. La liste $I(v_i)$ est donc composée de tuples de la forme $(v_j, H(X_{v_i,v_j}))$.

Les mesures locales sont ensuite échangées et fusionnées ente les nœuds du réseau pour déterminer les nœuds pathologiques. Cette fusion peut être conceptuellement représentée par une matrice $M_{entropy}$ contenant les mesures à l'échelle du réseau : chaque ligne et colonne représente un nœud du réseau. Si un nœud v_i mesure un nœud potentiellement pathologique v_j, une entrée $M_{entropy}[i, j]$ est définie dans la matrice et prend la valeur $H(X_{v_i,v_j})$. Cette entrée est vide dans le cas contraire. La matrice peut se limiter à une vue partielle du réseau ad-hoc. L'acquisition de cette matrice par un nœud local peut être réalisée en exploitant les mécanismes de diffusion d'un protocole de routage proactif. La matrice sera utilisée pour formuler les méthodes de détection.

7.4.2 Méthodes de détection collaboratives par seuil

Une première catégorie de méthodes distribuées repose sur une approche par seuil [140]. La détection consiste (1) à classer les nœuds intermittents présents dans le réseau ad-hoc selon un critère donné c, et ensuite (2) à identifier les nœuds pathologiques en fonction d'une valeur seuil λ (les nœuds sélectionnés sont ceux présentant une valeur $c(v_j)$ pour le critère c supérieure à λ). Nous proposons trois méthodes par seuil et décrivons chacune d'elles ci-dessous :

- la première méthode de détection m_1 (aussi appelée vote à la majorité) définit un classement des nœuds intermittents en fonction du nombre de nœuds observateurs qui ont considéré le nœud comme potentiellement pathologique. Pour un nœud ad-hoc v_j, le critère de classement correspond au nombre d'entrées non vides dans la matrice pour une colonne donnée $M_{entropy}[:, j]$. Il peut être défini par $c_1(v_j) = |\{M_{entropy}[i, j] \neq \emptyset\}|$. Un nœud sera détecté comme pathologique si et seulement s'il a été détecté localement comme pathologique par au moins λ_1 nœuds.
- la seconde méthode m_2 prend en compte le nombre de nœuds observateurs mais aussi la valeur de l'entropie mesurée par ces nœuds. Ainsi, m_2 classe les nœuds intermittents en fonction de la somme des valeurs d'entropie mesurées dans le réseau. Cette méthode est en fait une adaptation de la méthode m_1 où le résultat est pondéré par les valeurs d'entropie. Le critère correspond à la somme des valeurs en entrée d'une colonne donnée $M_{entropy}[:, j]$. Il est en conséquence défini par $c_2(v_j) = \sum_{i=1}^{n} M_{entropy}[i, j]$.
- la dernière méthode m_3 classe les nœuds intermittents en fonction de la valeur moyenne de l'entropie mesurée. m_3 ne s'intéresse pas au nombre de nœuds observateurs mais favorise la valeur de l'entropie mesurée en moyenne dans le réseau. Le troisième critère c_3 est en fait la valeur moyenne des entrées de la matrice dans une colonne donnée $M_{entropy}[:, j]$ et peut être formulée par : $c_3(v_j) = c_2(v_j)/c_1(v_j)$.

Ces méthodes peuvent être étendues en pondérant les mesures effectuées par les nœuds du réseau en fonction de leur fiabilité. Les mesures provenant de nœuds fiables auront des poids plus élevés et seront donc pris davantage en compte dans le processus de détection. La cohérence temporelle et la durée de vie des données de monitorage peuvent être améliorées en utilisant des approches telles que proposées dans [169] et [195]. Les caractéristiques de chacune des méthodes sont résumées dans le tableau 7.4.2. Les performances de ces méthodes seront évaluées par simulation dans la section 7.5.

Méthode	Critère
m_1 - Vote à la majorité	$c_1(v_j) = \|\{M_{entropy}[i, j] \neq \emptyset\}\|$
m_2 - Entropie cumulée	$c_2(v_j) = \sum_{i=1}^{n} M_{entropy}[i, j]$
m_3 - Entropie moyenne	$c_3(v_j) = c_2(v_j)/c_1(v_j)$

Tab. 7.1 – Récapitulatif des méthodes distribuées de détection par seuil

7.4.3 Auto-configuration avec la méthode des *k-means*

L'objectif de nos méthodes de détection consiste à classer les nœuds ad-hoc en deux populations distinctes : les nœuds intermittents sains et les nœuds intermittents pathologiques. Dans une méthode par seuil, le classement est réalisé en classant les nœuds intermittents et en sélectionnant les nœuds pathologiques à l'aide d'un seuil. Les performances de la méthode dépendent donc de la valeur seuil choisie. Nous définissons une solution auto-configurable alternative, capable de séparer les nœuds en deux classes de population sans avoir à paramétrer une

valeur seuil. Nous utilisons pour ce faire une méthode de classification par *k-means* [129] pour distinguer les nœuds en deux groupes.

La méthode est appliquée à l'ensemble des données obtenues par monitorage où l'on retrouve les différents nœuds intermittents. Un nœud intermittent v_j (colonne de la matrice $M_{entropy}$) est perçu comme un point dans un espace vectoriel à n dimensions, où une dimension correspond à la vue d'un nœud v_i (ligne de la matrice $M_{entropy}$). Les coordonnées $(x_1, .., x_n)$ d'un point v_j correspondent alors à $(M_{entropy[1,j]}, ...M_{entropy[n,j]})$. La norme d'un point v_a est notée $||v_a|| = \sqrt{\sum_{k=1}^{k=n}(M_{entropy}[k,a])^2}$. La distance entre deux points v_a et v_b vaut dans ce cas $||v_a - v_b|| = \sqrt{\sum_{k=1}^{k=n}(M_{entropy}[k,a] - M_{entropy}[k,b])^2}$. Intuitivement, la classification par *k-means* regroupe

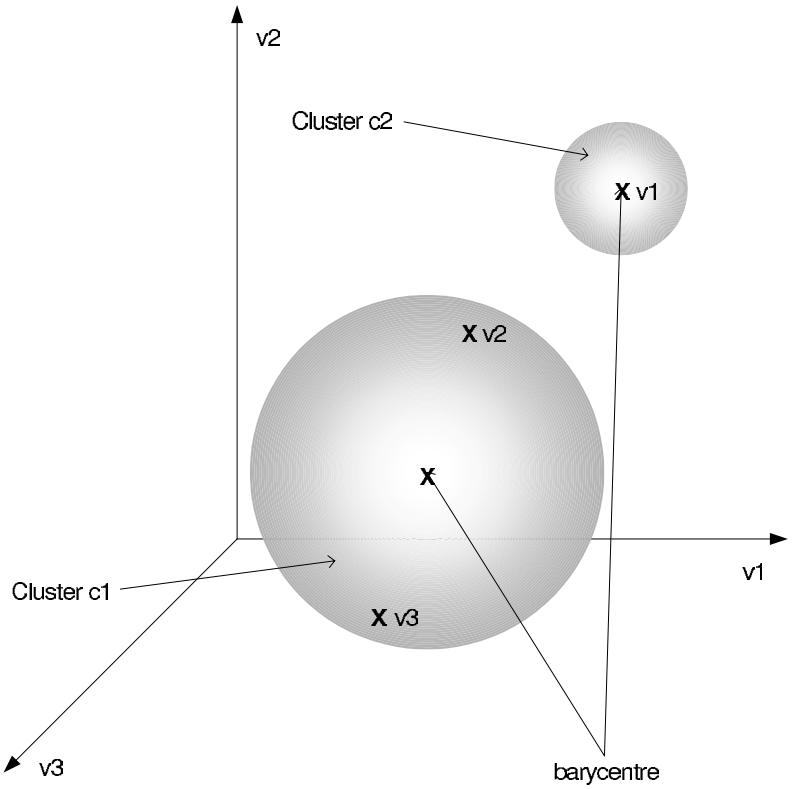


FIG. 7.7 – Application de la méthode des *k-means* dans un espace vectoriel à trois dimensions

les nœuds ad-hoc présentant des valeurs d'entropie similaires pour former deux groupes ou clusters. Le barycentre d'un cluster cl correspond au barycentre des points qui font partie de ce cluster dans l'espace vectoriel à n dimensions. L'algorithme des *k-means* est donné en 5. Il assigne chaque point v_j au cluster cl dont le barycentre (noté $barycentre(cl)$) est le plus proche. Les barycentres des clusters sont recalculés à chaque itération en fonction de la nouvelle répartition des points au sein des clusters. L'arrêt de l'algorithme est provoqué lorsque les barycentres restent inchangés. L'algorithme consiste dans les faits à minimiser la fonction de coût définie par l'équation 7.10.

$$E = \frac{1}{2} \sum_{cl \in \{cl_1, cl_2\}} \sum_{v_j \in cl} ||v_j - barycentre(cl)||^2 \tag{7.10}$$

Dans cette équation, cl_1 et cl_2 sont les clusters représentant les deux populations de nœuds et $||v_j - barycentre(cl)||$ est la distance entre un point v_j et le barycentre de son cluster cl.

Les nœuds considérés comme pathologiques sont regroupés dans le cluster dont le barycentre a la norme la plus élevée (entropie la plus élevée). La figure 7.7 décrit un exemple d'application de la méthode dans un espace vectoriel à trois dimensions où les trois points v_1, v_2 et v_3 sont répartis en deux clusters.

Algorithme 5 : Auto-configuration avec la méthode des *k-means*

Données : Matrice $M_{entropy}$ contenant les mesures d'intermittence
Résultat : Ensemble des nœuds considérés pathologiques
initialisation
a) sélectionner deux clusters $\{cl_1, cl_2\}$ arbitrairement ;
b) calculer les barycentres initiaux $barycentre(cl_1)$ et $barycentre(cl_2)$;
répétition
c) assigner ou réassigner chaque point v_j au cluster dont le barycentre est le plus proche du point considéré ;
d) calculer les deux nouveaux barycentres correspondant à chacun des clusters ;
jusqu'à
les barycentres $barycentre(cl_1)$ et $barycentre(cl_2)$ restent inchangés ;
enfin
les nœuds considérés pathologiques sont présents dans le cluster (cl_1 ou cl_2) dont le barycentre a la norme la plus élevée.

Cette auto-configuration permet de détecter les nœuds pathologiques sans nécessiter de fixer une valeur seuil à proprement parler. La méthode des *k-means* offre à la fois facilité de déploiement et flexibilité. Nous verrons dans la section suivante qu'elle nous a permis d'aboutir à des résultats satisfaisants en moyenne comparativement aux méthodes de détection par seuil. Il est à noter que la convergence de l'algorithme est prouvée [174] et qu'une analyse des propriétés de convergence est disponible dans [174]. Cependant, la méthode des *k-means* ne garantit pas de trouver systématiquement le minimum global de la fonction de coût E que la classification des *k-means* cherche à minimiser. Elle peut notamment converger vers des optimums locaux en fonction des barycentres initiaux. Plusieurs solutions algorithmiques énoncées dans [201] permettent de garantir le minimum global. Il est donc encore possible d'améliorer les performances de cette méthode d'auto-configuration, déjà satisfaisante dans notre contexte.

7.5 Résultats expérimentaux

Les performances de notre approche ont été expérimentées à travers un ensemble de simulations. Ceci nous a permis d'évaluer les différentes méthodes de détection et d'estimer l'impact du modèle de mobilité et du modèle de fautes. Les expérimentations ont été effectuées avec ns-2 [74]. Les paramètres de simulation sont récapitulés dans le tableau 7.2. Pour chaque expérimentation, un ensemble de nœuds pathologiques (de 0 à 5 nœuds) est choisi aléatoirement et suit la chaîne de Markov à deux états (voir la figure 7.3.2) avec les probabilités de transition (p, q). Cet ensemble de nœuds pathologiques est ensuite comparé à l'ensemble des nœuds détectés comme pathologiques par la méthode de détection.

Sensibilité et spécificité

Afin de quantifier les performances de notre approche, nous avons réalisé une analyse de la sensibilité et de la spécificité des méthodes de détection [8]. Notre approche peut être vue

Paramètre	Valeur
Simulateur	ns-2
Durée de simulation	900 s
Surface de simulation	1500 m x 300 m
Nombre de nœuds ad-hoc	50 nœuds
Nombre de nœuds pathologiques	0 - 5 nœud(s)
Modèle de mobilité	Random WayPoint *mobgen - steady state*
Vitesse	0 - 10 m/s
Temps de pause	0 - 120 s
Couche MAC	IEEE 802.11
Couche routage	NRL OLSR

TAB. 7.2 – Paramètres de simulation

comme un test diagnostic, où nous testons si un nœud ad-hoc est pathologique (test positif) ou sain (test négatif). En comparant l'ensemble des nœuds initialisés comme pathologiques à celui des nœuds dont le test est positif, nous pouvons déterminer si le test fournit un résultat vrai ou un résultat faux. Par exemple, un vrai positif (*TP, True Positive*) signifie que le test a réussi à détecter un nœud pathologique tandis qu'un faux positif (*FP, False Positive*) signifie que le test a échoué. Nous définissons également la notion de vrai négatif (*TN, True Negative*) et de faux négatif (*FN, False Negative*) pour la détection d'un nœud sain. Pour déterminer l'intérêt et l'efficacité de la méthode, nous utilisons les facteurs de sensibilité (*Sn*) et de spécificité (*Sp*) définis dans l'équation 7.11.

$$Sn = \frac{TP}{TP + FN} \qquad Sp = \frac{TN}{TN + FP} \tag{7.11}$$

La sensibilité révèle la capacité du test diagnostic à détecter les cas pathologiques : elle correspond à la proportion de tests positifs parmi tous les tests où les nœuds étaient pathologiques. La spécificité quant à elle révèle la capacité du test diagnostic à détecter les cas sains : elle correspond à la proportion de tests négatifs parmi tous les tests où les nœuds étaient sains. Pendant une méthode de détection par seuil, la matrice $M_{entropy}$ est seuillée pour déterminer les nœuds pathologiques. Le test diagnostic offre un compromis entre sensibilité et spécificité. Une idée intuitive de ce compromis peut être obtenue en analysant deux cas extrêmes en termes de valeurs seuil. Dans un premier cas extrême, la méthode détecte tous les nœuds comme pathologiques (valeur seuil fixée à zéro). Dans ce cas, la sensibilité est maximale (valeur de 1) alors que la spécificité est minimale (valeur nulle). Dans un second cas extrême, la méthode détecte tous les nœuds comme sains (valeur fixée à l'infini). Dans ce cas, la sensibilité est minimale (valeur nulle) alors que la spécificité est maximale (valeur de 1).

Représentation graphique ROC

Nous utilisons une représentation graphique appelée ROC (*Receiver Operating Characteristic*) [204] qui correspond à la comparaison de la sensibilité (*Sn*) par rapport à 1-spécificité (*1 - Sp*) permettant de rendre compte de ce compromis lorsque l'on souhaite comparer la performance d'un test diagnostic. Une méthode de détection idéale présente une courbe ROC qui se situe dans la partie en haut à gauche du graphe, puisque la sensibilité (tous les tests positifs sont vrais) et la spécificité (tous les tests négatifs sont vrais) atteignent tous les deux des valeurs

maximales. Si l'on délimite le graphique en deux parties séparées par une ligne imaginaire d'un angle de 45 degrés allant du coin en bas à gauche au coin en haut à droite du graphe, le test devient inefficace lorsque la courbe se situe dans la première moitié en bas à gauche. En effet, le taux de faux positifs devient supérieur au taux de vrais positifs : le test a alors une plus grande probabilité de considérer un nœud pathologique comme sain que comme pathologique.

Dans les prochaines sous-sections, nous allons détailler les résultats expérimentaux (1) en présentant et analysant les courbes ROC pour comparer les performances des trois méthodes de détection par seuil, (2) en évaluant l'impact du modèle de mobilité (paramètres (*temps_pause*, *vitesse_max*) de RWP) et du modèle de fautes (paramètres (p, q) de la chaîne de Markov) sur notre approche et (3) enfin en déterminant les performances de la méthode d'auto-configuration.

7.5.1 Performances comparées des méthodes de détection

Nous avons analysé avec une première série d'expériences les performances des trois méthodes de détection par seuil décrites dans la section 7.4.2. Ces résultats sont présentés sur la figure 7.8 et reposent sur un ensemble étendu de simulations incluant différents paramètres de mobilité $(temps\_pause, vitesse\_max)$ et paramètres de fautes (p, q). Nous avons considéré un temps de pause variant de 0 à 120 secondes et une vitesse maximale variant de 0.1 à 10 m/s pour la mobilité. Les nœuds pathologiques sont paramétrés avec des probabilités de transition réalistes. La probabilité de tomber en panne p est configurée avec une valeur faible de 0.1 à 0.2 et la probabilité d'être réparé q avec une valeur comprise entre 0.1 et 1.0. Pour chaque configuration, nous avons réalisé 150 simulations pour éviter le biais des résultats. Les performances des

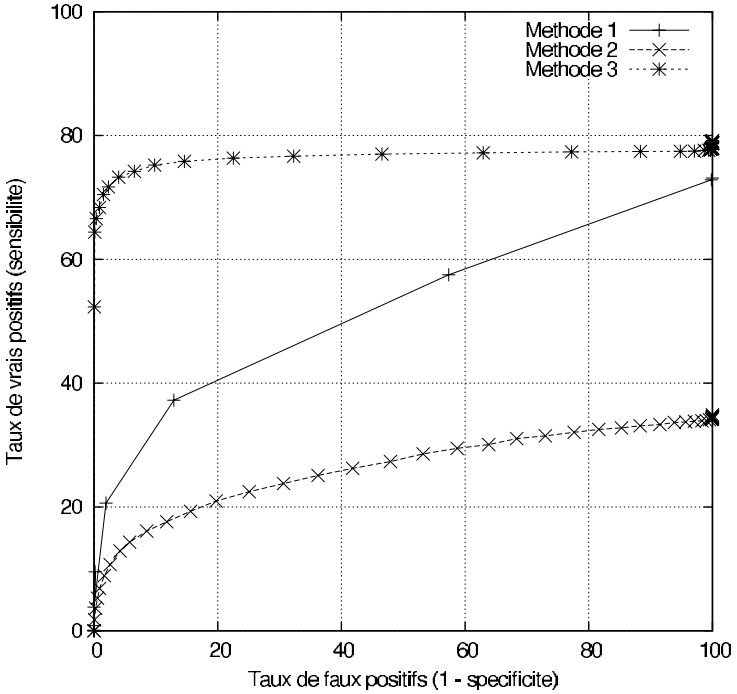


Fig. 7.8 – Courbes ROC pour les trois méthodes par seuillage

méthodes de détection sont synthétisées sur la figure 7.8, où nous avons tracé la courbe ROC pour chacune des méthodes. Un point (x,y) sur la courbe représente le taux de vrais positifs (y) comparé au taux de faux positifs (x) de la méthode pour une valeur seuil donnée. Nous sommes

intéressés par une méthode performante qui fournit un taux faible de faux positifs pour un taux maximal de vrais positifs. Plus la méthode se situe dans le coin supérieur gauche de l'espace ROC, plus elle fournit une détection efficace.

Nous pouvons donc déduire graphiquement que la méthode m_3 qui repose sur la valeur moyenne de l'entropie représente un test diagnostic meilleur que les deux autres méthodes. En particulier, la méthode m_3 offre de bons résultats avec un taux de vrais positifs supérieur à 75% dans la plupart des cas. De manière plus fine, si l'on souhaite un taux de faux positifs inférieur à 20%, la méthode m_3 avec plus de 75% de vrais positifs est définitivement meilleure que la méthode m_1 avec un taux de vrais positifs de moins de 45%, et bien meilleure encore que la méthode m_2 avec un taux de vrais positifs de moins de 20%. Il s'avère que les méthodes m_1 et m_2 présentent des performances moins convaincantes. Cela provient notamment du fait que la détection est trop dépendante du nombre de nœuds observant le nœud intermittent. Par exemple pour la méthode m_1, la détection repose sur le vote à la majorité et par conséquent la probabilité d'un nœud ad-hoc d'être considéré comme pathologique croît avec le nombre de nœuds qui est présent dans son voisinage. De la même manière, la méthode m_2 considère la valeur cumulée de l'entropie à l'échelle du réseau. Ce critère dépend également du nombre de nœuds voisins. Avec la méthode m_3, la valeur moyenne de l'entropie fournit un critère plus indépendant et fiable pour mesurer l'intermittence, où une augmentation du nombre de nœuds voisins permet d'améliorer et de raffiner la valeur moyenne de l'entropie sans dénaturer la mesure.

7.5.2 Impact du modèle de mobilité sur le monitorage

Une seconde série d'expériences portait sur l'impact de la mobilité sur la détection de nœuds. Nous voulions quantifier le fait que les performances se dégradent lorsque la mobilité augmente : l'entropie générée par la mobilité augmentant et ne permettant plus de distinguer celle générée par la pathologie. Nous avons considéré pour ce faire différents paramètres de mobilité avec un modèle d'intermittence réaliste (paramètres $p = 0.1$ et $q = 0.4$). Nous avons configuré les paramètres de mobilité avec un temps de pause raisonnable de 0 à 120 secondes et une vitesse maximale comprise entre 1 et 10 m/s. Nous avons ensuite mesuré la sensibilité et la spécificité de la méthode m_3, qui s'était avérée la plus efficace comparativement aux deux autres dans les expériences précédentes. Ces résultats sont présentés sur la figure 7.9 où nous avons tracé les courbes ROC pour chaque couple $(temps\_pause, vitesse\_max)$ de paramètres de mobilité. Nous étudions les performances de la méthode de détection pour des configurations avec un taux de faux positifs relativement bas. Nous nous sommes limités au tracé des courbes ROC pour un taux de faux positifs inférieur à 20%. La comparaison des courbes ROC montre que l'impact de la mobilité est relativement limité pour des scénarios réalistes. La différence de performances entre la mobilité la plus faible et la mobilité la plus élevée est en effet de moins de 5%. Ce constat vient de la nature de notre mesure qui met davantage en évidence l'entropie additionnelle générée par une pathologie plutôt que celle induite par la mobilité.

Une mobilité très élevée génère de mauvais résultats où par mauvais résultats nous entendons un faible taux de vrais positifs pour un faible taux de faux positifs de moins de 20%. En effet, les performances sont fortement dégradées dans des scénarios, comme par exemple $temps\_pause = 0$ s et $vitesse\_max = 10$ m/s, avec une sensibilité de moins de 72%. En revanche, une mobilité très faible peut également s'avérer problématique. En réalité, l'impact de la mobilité est double. Dans un premier temps, la détection est améliorée lorsque la mobilité augmente pour les paramètres de mobilité de $(120, 1)$ à $(30, 10)$. La mobilité augmente le nombre de nœuds observateurs impliqués dans la détection et permet ainsi de raffiner les mesures d'intermittence. Dans un second temps, la détection est perturbée avec les scénarios de forte mobilité. Elle n'est plus capable de mettre

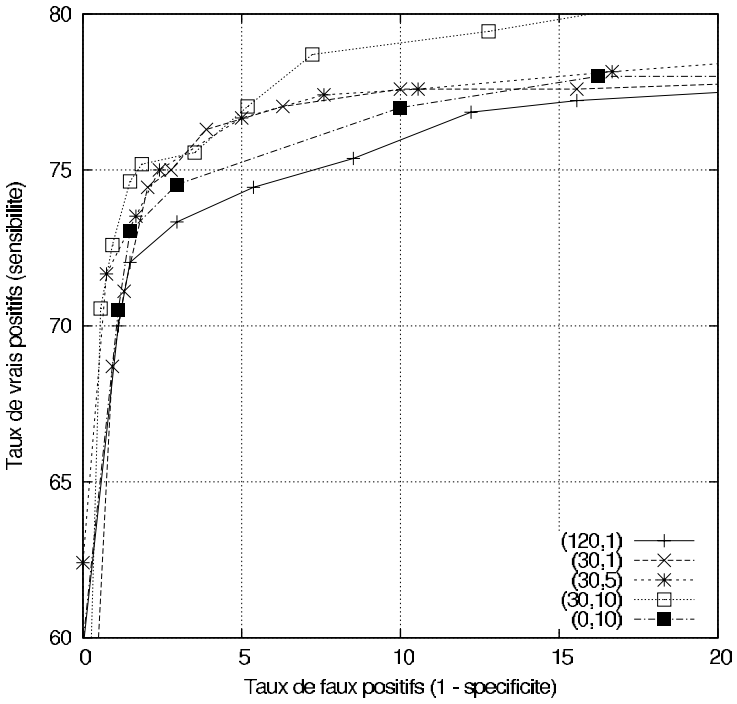


FIG. 7.9 – Impact du modèle de mobilité sur la méthode de détection par seuil m_3 [A partir d'un ensemble de simulations, nous avons tracé la courbe ROC pour la méthode de détection m_3 avec différents paramètres de mobilité $(temps\_pause, vitesse\_max)$.]

en évidence efficacement les nœuds pathologiques. En bref, la détection montre de meilleurs résultats lorsque la mobilité ne correspond pas à des scénarios extrêmes (très faible et très forte mobilité).

7.5.3 Impact du modèle de fautes sur le monitorage

Dans une troisième série d'expériences, nous avons analysé dans quelle mesure le modèle d'intermittence a un impact sur la méthode de détection m_3. Nous avons considéré un modèle de mobilité avec des paramètres réalistes pour le modèle d'intermittence : valeur p constante de 0.1 et valeur q variant de 0.1 à 0.9. Nous avons mesuré, avec la même approche que précédemment, la sensibilité et la spécificité de la méthode m_3 fondée sur l'entropie moyenne. Les résultats sont présentés sur la figure 7.10 où une courbe est tracée pour différents paramètres p et q. La différence de performances entre les pires et les meilleurs résultats atteint la valeur de 8%. Nous pouvons observer que cette variation est plus significative que lorsque nous avons évalué les différentes mobilités. Elle confirme l'idée que notre approche est capable de caractériser l'intermittence anormale de nœuds pathologiques en mettant en évidence l'entropie générée additionnellement. A partir de l'analyse théorique, nous avons montré que l'entropie $H(X_{v_i,v_j})$ décroît pour une probabilité p donnée lorsque la probabilité de réparation q augmente $(q > p)$. Sur la figure 7.10, la sensibilité de la détection traduit le même comportement : la sensibilité décroît lorsque la probabilité q augmente. Notre méthode de détection est fondée sur les mesures d'entropie. Lorsque l'entropie $H(X_{v_i,v_j})$ est réduite, la méthode de détection est détériorée et la sensibilité diminue en conséquence. Sensibilité et entropie sont corrélées positivement, les résultats expérimentaux confirment donc notre analyse théorique.

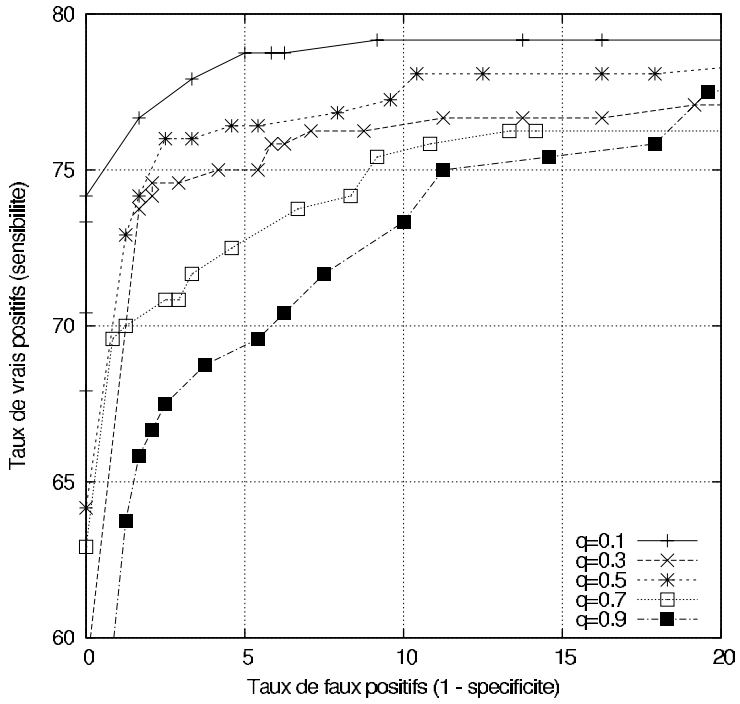


FIG. 7.10 – Impact du modèle de fautes sur la méthode de détection par seuil m_3 [Nous avons tracé la courbe ROC de la méthode de détection m_3 avec différents paramètres p et q]

Performance de la méthode de détection auto-configurable

Dans une quatrième série d'expériences, nous avons décidé d'appliquer la même évaluation de performances (analyse de l'impact de la mobilité et de l'impact de l'intermittence) pour la méthode auto-configurable utilisant l'algorithme de classification *k-means* [129]. Les même paramètres que précédemment ont été considérés durant ces simulations pour le modèle de mobilité et le modèle de fautes. Cependant, la représentation sous forme de courbes ROC n'a plus d'intérêt dans ce cas précis puisque la méthode ne requière aucune valeur seuil. Nous avons donc limité notre analyse des simulations au tracé des courbes présentées en figures 7.11 et 7.12.

La première figure présente la sensibilité de la méthode auto-configurable pour différents paramètres de mobilité et confirme le même comportement que celui constaté avec les méthodes par seuil : la sensibilité croît dans une première étape puis décroît lorsque la mobilité devient plus importante. La deuxième figure décrit l'impact du modèle de fautes et montre que les résultats de simulations sont cohérents avec l'analyse de la méthode m_3 : la sensibilité de la méthode décroît lorsque la probabilité de réparation q augmente. La méthode auto-configurable fournit une valeur de sensibilité unique alors qu'une méthode par seuil aboutit à un intervalle de valeurs de sensibilité lorsque l'on parcourt les différentes valeurs seuils. Pour comparer la sensibilité obtenue avec la méthode auto-configurable et celle obtenue avec la meilleure méthode par seuil m_3, il est nécessaire de comparer les valeurs des courbes de sensibilité présentées aux figures 7.11 et 7.12 avec les intervalles de valeurs de sensibilité des courbes ROC présentées aux figures 7.9 et 7.10. On constate alors que la méthode auto-configurable n'offre pas de meilleures performances comparées à la borne supérieure des valeurs de sensibilité obtenues avec la méthode par seuil m_3. En revanche d'un point de vue plus réaliste, la configuration d'une valeur de seuil implique que celle-ci ne soit pas *a priori* optimale, puisqu'elle peut évoluer en fonction des

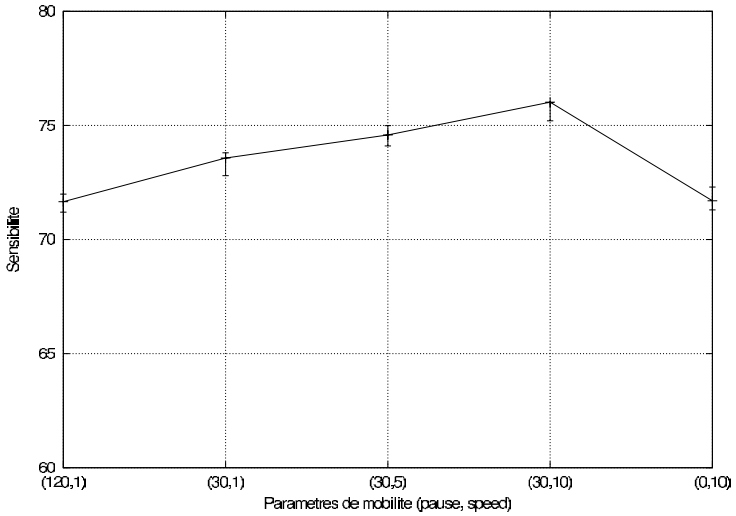


FIG. 7.11 – Impact du modèle de mobilité sur la méthode auto-configurable

caractéristiques du réseau ad-hoc. La sensibilité de la méthode auto-configurable doit donc être comparée à la moyenne des valeurs de sensibilité de la méthode par seuil. Dans ce cas, on constate que les performances sont équivalentes. Par conséquent, la méthode auto-configurable sera privilégiée puisqu'elle ne nécessite aucun paramétrage spécifique et elle fournit globalement des performances de détection plus stables.

7.6 Synthèse

Nous proposons une approche de gestion de fautes reposant sur une analyse de l'intermittence des nœuds au sein d'un réseau ad-hoc. L'intermittence d'un nœud peut être provoquée par des causes bénignes telles que la mobilité ou la dégradation temporaire de la connectivité. Elle est cependant pathologique lorsqu'elle est causée par des pannes de routage, des problèmes de batterie ou des perturbations physiques fortes.

Cette approche de gestion repose sur trois concepts clé : (1) une mesure fondée sur la théorie de l'information permettant localement d'identifier les nœuds intermittents dans le plan de routage, (2) des méthodes de détection collaboratives entre nœuds permettant de détecter les comportements pathologiques dans le réseau de manière distribuée et (3) un mécanisme d'auto-configuration utilisant l'algorithme de classification des *k-means*.

La solution a été expérimentée à travers un ensemble de simulations qui a permis de comparer les performances des différentes méthodes de détection et de quantifier l'impact de la mobilité et du modèle de fautes. Nous avons montré comment la corrélation des données de monitorage en provenance des différents nœuds aboutit à une détection plus efficace et plus fiable des pathologies.

Les principaux avantages de notre approche de gestion de fautes sont : son faible coût de fonctionnement puisqu'elle s'appuie sur une analyse du plan de routage, sa robustesse puisqu'elle met en œuvre des méthodes de détection distribuée collaborative et sa capacité à détecter des nœuds pathologiques non instrumentés puisque que la méthode est opérée de manière indirecte par des nœuds observateurs.

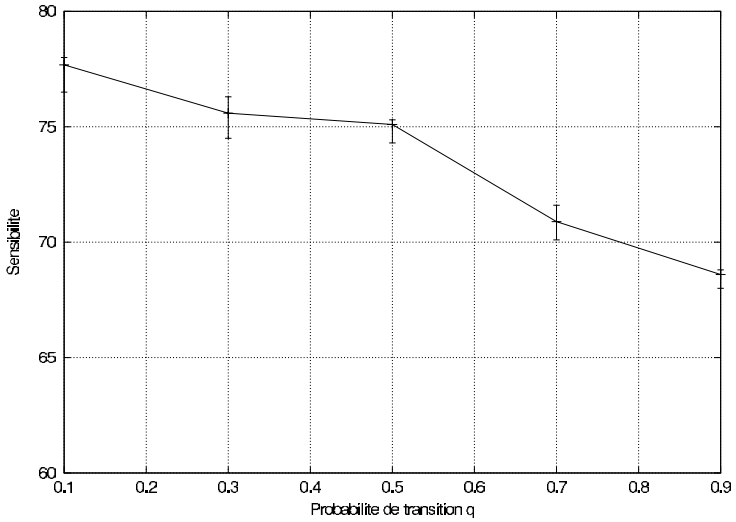


FIG. 7.12 – Impact du modèle de fautes sur la méthode auto-configurable

Les travaux présentés dans ce chapitre ont fait l'objet de plusieurs publications [17, 16].

Troisième partie

Mise en œuvre

Chapitre 8

Développement d'une plate-forme de monitorage

Sommaire

8.1 Introduction

Ce chapitre présente la mise en œuvre des différentes contributions à travers le développement d'une plate-forme de monitorage. Le monitorage est une activité d'observation qui nous permet d'évaluer l'état opérationnel et le fonctionnement d'un réseau ad-hoc. L'activité comprend la mesure, la collecte, l'analyse ainsi que le stockage des données statistiques incluant paramètres et mesures de performance. L'objectif de cette mise en œuvre est de compléter la validation de nos travaux de recherche par la pratique, en complément des travaux de simulations. Le développement de la plate-forme provient pour l'essentiel de deux projets de recherche avec des étudiants en école d'ingénieur à ESIAL. Mickaël Chaffangeon, Sandra Reichert et Pierre-Yves Thomas ont travaillé sur le prototypage de la plate-forme en s'intéressant plus particulièrement au développement d'une sonde ad-hoc [51]. Salifou Mahaman et Jean-Claude Tranquillin ont complété l'implantation par l'ajout d'un module d'analyses statistiques [130]. L'utilisation de patrons de conception [88] a largement été encouragé dans le cadre de ces réalisations pour offrir une infrastructure facilement extensible. Après avoir présenté un aperçu de la plate-forme de monitorage, nous détaillerons l'architecture fonctionnelle du prototype et les fonctionnalités du module d'analyses statistiques que nous illustrerons à travers un ensemble de cas d'utilisation.

8.2 Plate-forme de monitorage

Le développement de la plate-forme de monitorage présente un double enjeu. Elle permet tout d'abord d'implanter notre modèle d'information présenté au chapitre 2. Ce modèle offre un cadre formel pour la description des ressources et la structuration de l'information de gestion dans le contexte des réseaux et services ad-hoc. Sa spécification se présente sous la forme d'une extension du modèle commun de l'information (CIM) [56]. Notre modèle d'information de gestion doit être suffisamment générique pour être applicable à toute forme de scénarios. En particulier, il est nécessaire de vérifier sa correcte instanciation à travers une implantation concrète.

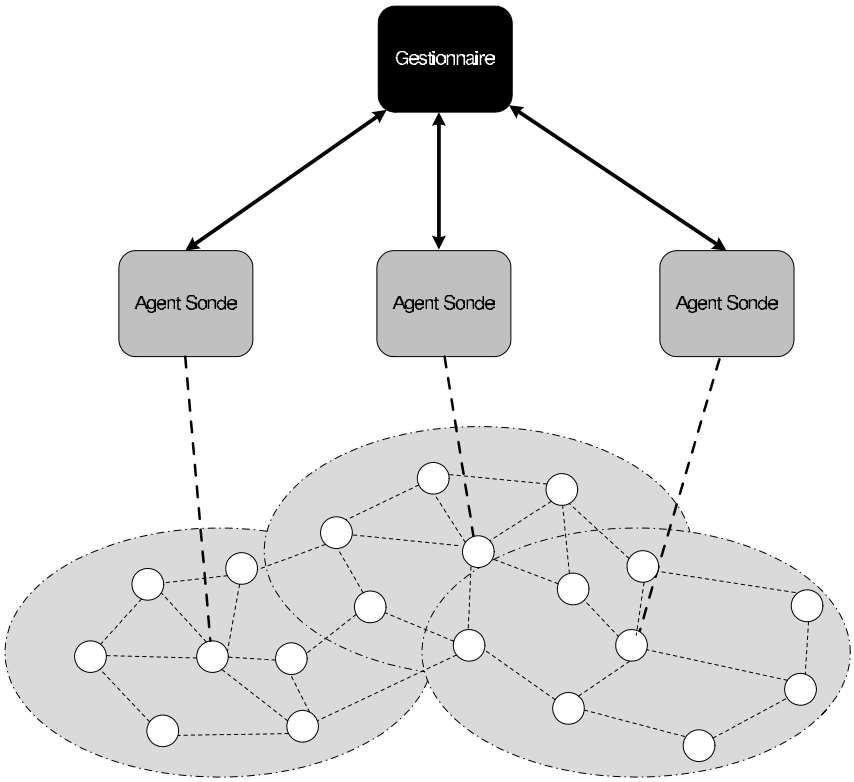


Fig. 8.1 – Vue générale de la plate-forme de monitorage pour les réseaux ad-hoc

D'autre part, cette mise en œuvre permet d'expérimenter les différentes techniques d'analyse que nous avons utilisées pour organiser le plan de supervision et adapter les opérations de gestion. Des filtres de similitude et de différence ont été employés à des fins d'analyse de performance. L'analyse de la centralité par vecteur propre nous a permis d'identifier les nœuds importants et l'étude de la distribution des paquets dans le plan de routage de détecter les nœuds pathologiques. L'implantation de la plate-forme permet d'appliquer ces méthodes dans un environnement réel et ainsi de compléter les expériences réalisées par la simulation.

Une vue générale de notre plate-forme de monitorage est décrite à la figure 8.1 : elle dispose d'une structure relativement légère composée d'un gestionnaire principal et d'un sous-ensemble d'agents-sondes implantés parmi les nœuds du réseau ad-hoc. Tous les nœuds du réseau ne sont donc pas instrumentés, seuls les agents-sondes sont intégrés au plan de gestion et ont la charge d'observer l'activité des nœuds situés dans leur voisinage. Techniquement parlant, chaque agent-sonde dispose d'une carte réseau en mode promiscuité lui permettant d'observer toutes les trames transmises dans son voisinage quelqu'en soit la destination. Les données de monitorage

sont ensuite collectées par le gestionnaire principal pour offrir une vue synthétique du réseau ad-hoc. Le monitorage est réalisé de manière passive par les agents-sondes, sans nécessiter l'injection de paquets spécifiques sur le réseau ad-hoc. L'activité consiste à observer et analyser directement en temps réel le trafic de paquets qui transite au sein du réseau.

Comparativement à l'outil de monitorage WANMON (*Wireless Ad-Hoc Network Monitoring Tool*) [149] décrit précédemment, nous constatons un certain nombre de similarités : le déploiement d'un agent local permettant d'analyser le trafic réseau, ainsi que la possibilité d'évaluer la participation d'un nœud en distinguant la part du trafic générée par le nœud en tant que terminal de celle générée par le nœud en tant que routeur. Si l'outil est capable de déterminer statistiquement le coût du routage de manière approfondie en termes de trafic, de consommation d'énergie, d'occupation mémoire et de charge CPU, le monitorage en tant que tel se limite à analyser le comportement du nœud local. L'agent-sonde permet en revanche d'observer le comportement de l'ensemble des nœuds ad-hoc localisé dans le voisinage proche en sondant leur trafic réseau.

8.3 Prototype de la plate-forme

La première étape de notre travail a consisté à développer un prototype de la plate-forme qui dispose d'un minimum de fonctionnalités. Nous entendons par minimum de fonctionnalités la capacité d'analyser sommairement les paquets échangés et de construire la topologie du réseau. La figure 8.2 présente l'architecture fonctionnelle du prototype : ce dernier est composé d'un gestionnaire principal ainsi que d'un agent-sonde.

La configuration choisie permet une expérimentation de la plate-forme de bout en bout : de la capture des trames par l'agent-sonde à l'interface graphique du gestionnaire offrant une vue synthétique du réseau ad-hoc. Le gestionnaire et l'agent-sonde peuvent être déployés sur un même nœud ad-hoc dans le scénario le plus simple. Cependant, l'architecture doit permettre de répartir les deux composants sur deux nœuds distincts en permettant des échanges distants. Nous allons détailler le fonctionnement de chacun des composants puis en décrire les différentes interactions.

L'essentiel du prototype de la plate-forme a été réalisé avec le langage de programmation Java [77]. Le choix de ce langage est motivé par plusieurs raisons. De part sa nature orientée objet, ce langage permet une transcription aisée des classes de notre modèle d'information. Il offre par ailleurs une grande souplesse en terme de déploiement et dispose d'importantes bibliothèques de classes qui permettront un développement efficace de l'interface graphique du gestionnaire et du module d'analyses statistiques. La capture des trames s'appuie en revanche sur la bibliothèque de fonctions *libpcap* [6] développée en C [121] pour des raisons évidentes de performance.

8.3.1 Composant agent-sonde

Le composant agent-sonde a pour fonction première la capture des paquets circulant au sein du réseau ad-hoc. Son déploiement est réalisé sur un équipement mobile disposant d'une carte sans fil configurée en mode promiscuité. Par défaut, une carte sans fil détruit immédiatement les trames qui ne lui sont pas destinées pour réduire la charge de traitements. Afin d'observer le voisinage proche, nous configurons la carte avec le mode promiscuité qui permet de récupérer l'intégralité du trafic réseau sans condition sur le destinataire de la trame.

La bibliothèque de fonctions *libpcap* est utilisée pour capturer les paquets. De nombreux outils réseaux tels qu'*ethereal* [59] et *snort* [168] exploitent cette célèbre bibliothèque car elle fournit une interface de programmation complète. Elle permet d'effectuer des captures à partir de

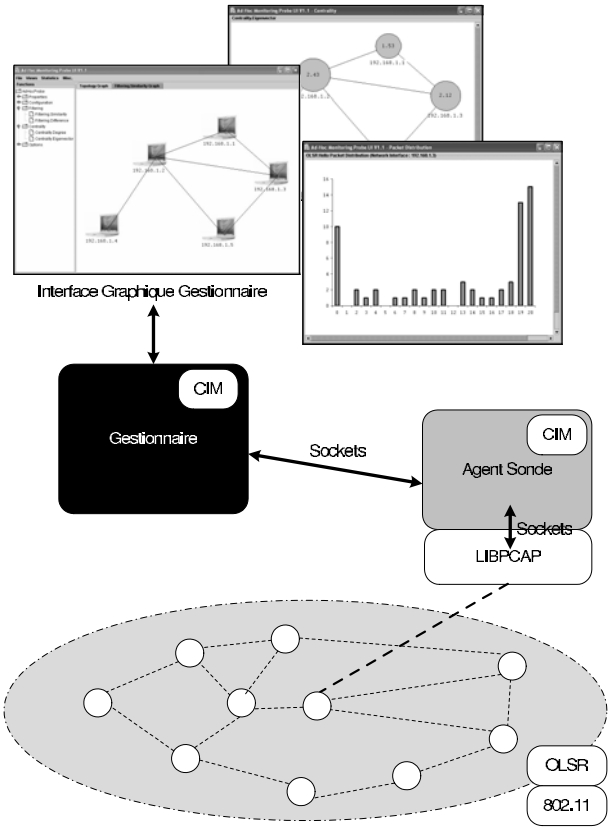


FIG. 8.2 – Architecture fonctionnelle de la plate-forme avec un agent-sonde

différents supports, et dispose d'un mécanisme de filtrage intégré utilisant BPF (*Berkeley Packet Filtering*) [136] implanté au sein du noyau système pour offrir de meilleures performances.

Le corps de l'agent sonde est développé en Java pour faciliter notamment l'implantation de la base d'informations. Les échanges avec la bibliothèque *libpcap* développée en C reposent sur l'utilisation de sockets. L'agent-sonde capture les paquets à partir de cette bibliothèque et maintient parallèlement une base d'informations sur le trafic réseau observé. La base d'informations est ensuite mise à la disposition du composant gestionnaire.

8.3.2 Composant gestionnaire

Le composant gestionnaire est chargé de configurer le composant agent-sonde et de collecter les données de monitorage auprès de celui-ci afin de construire une vue synthétique du réseau ad-hoc. La collecte correspond à des requêtes périodiques du gestionnaire à l'agent-sonde. Le gestionnaire dispose d'une interface graphique permettant d'interagir avec l'utilisateur et de fournir des informations statistiques sur le réseau.

Le gestionnaire est entièrement réalisé en Java et implante le patron de conception Modèle-Vue-Contrôleur. Le modèle représente la base d'informations de gestion du gestionnaire tandis que la vue constitue l'interface graphique fournie à l'utilisateur. Le principe de ce patron est relativement intuitif : lorsqu'un utilisateur émet une requête par le biais de l'interface graphique, celle-ci est analysée par le contrôleur. Ce dernier effectue les traitements appropriés auprès du modèle, en l'occurence la base d'informations de gestion. La modification du modèle provoque la mise à jour de la vue, donc de l'interface graphique du gestionnaire.

8.3.3 Interactions entre composants

Les interactions entre les différents composants : gestionnaire, agent-sonde et module *libpcap* sont représentées par le diagramme de séquence à la figure 8.3. Le module *libpcap* représente en fait un programme que nous avons développé en C intervenant comme interface pour interroger la bibliothèque *libpcap*. Les échanges entre le composant gestionnaire et le composant agent-

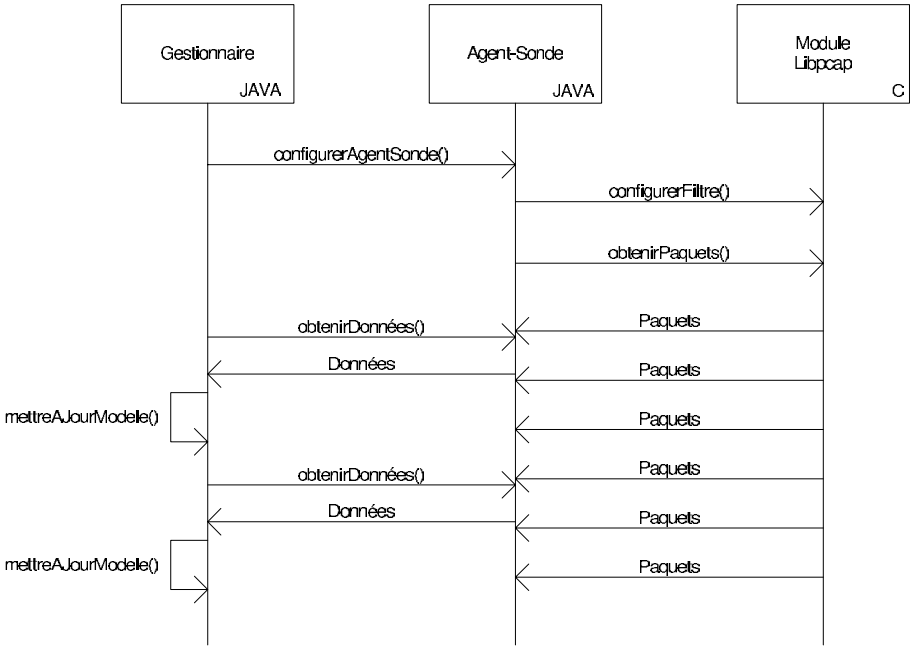


FIG. 8.3 – Diagramme de séquences UML présentant les interactions entre le composant gestionnaire, le composant agent-sonde et le module *libpcap*

sonde sont opérés à l'aide de sockets pour assurer les communications distantes. Ils permettent au gestionnaire de configurer l'agent-sonde et d'opérer la collecte des données de gestion en interrogeant la base d'informations de l'agent. L'agent-sonde et le module *libpcap* sont déployés sur un même nœud local mais échangent à l'aide de sockets car leurs implantations correspondent à deux langages de programmation différents. L'agent-sonde peut configurer le mécanisme de filtrage intégré de *libpcap* et recevoir les différents paquets capturés. Les bases d'informations du gestionnaire et de l'agent-sonde sont mises à jour régulièrement. Bien que n'étant pas représentée explicitement sur le diagramme pour des raisons de visibilité, la mise à jour de la base d'informations de l'agent-sonde est faite périodiquement à la réception de nouveaux paquets.

8.3.4 Fonctionnalités de base

Afin d'évaluer le bon fonctionnement du prototype de la plate-forme, nous avons intégré un ensemble de fonctionnalités de base qui permet d'identifier les caractéristiques élémentaires d'un réseau ad-hoc.

La première fonctionnalité permet de déterminer les nœuds présents dans le réseau et d'analyser sommairement les paquets émis par ceux-ci. Les paquets capturés par le composant agent-sonde sont triés et comparés en fonction des entêtes MAC 802.11 et des entêtes IP. La présence de nouveaux nœuds est identifiée en observant les adresses sources des paquets capturés. Le nombre de trames et de paquets IP émis par un nœud ou à destination d'un nœud est inventorié

dans la base d'informations de gestion.

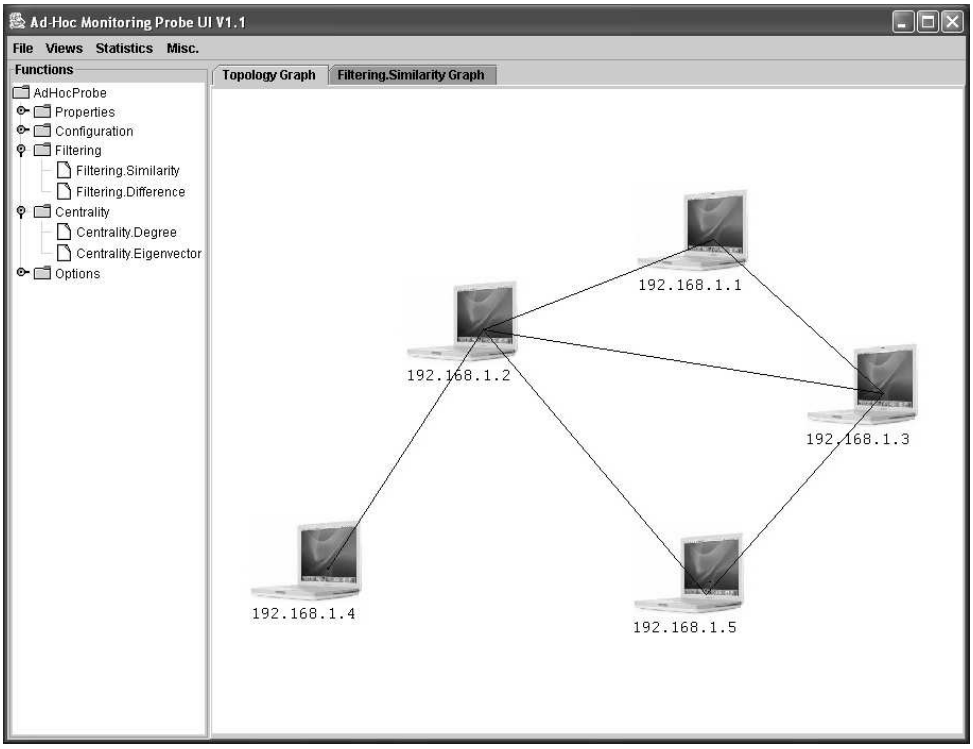


FIG. 8.4 – Interface graphique fournie par le gestionnaire

La seconde fonctionnalité consiste à établir la topologie partielle du réseau en corrélant les adresses sources et destinations des paquets. L'échange de nombreuses trames entre deux nœuds ad-hoc permet d'identifier l'existence de liens physiques d'une durée de vie significative. La figure 8.4 décrit l'interface graphique offerte par le gestionnaire : cette dernière présente la construction d'une topologie d'un réseau ad-hoc obtenue dans le cadre de nos expérimentations.

La dernière fonctionnalité porte sur la détection des nœuds OLSR. Une partie entière de notre modèle d'information est dédiée à ce protocole de routage et une part importante de nos méthodes de gestion s'appuie sur celui-ci. Un nœud OLSR est caractérisé par l'émission de paquets de contrôles spécifiques, qui lui permet d'identifier le voisinage et de transmettre les informations de topologie. L'identification de paquets de contrôle OLSR permet de détecter les nœuds implantant le protocole.

8.4 Module d'analyses statistiques

La deuxième étape de notre travail portait sur le développement d'un module d'analyses statistiques, en complément des fonctionnalités de base de notre plate-forme. Ce module est intégré au composant gestionnaire afin d'appliquer des traitements sur la base d'informations de gestion. Il nous permet de mettre en œuvre notamment les méthodes à base de filtres de similitude et de différence, l'analyse de la centralité par vecteur propre et l'étude de la distribution de paquets décrites précédemment.

8.4.1 Distributions de paquets

Le module permet tout d'abord d'analyser statistiquement la distribution des paquets dans le temps. L'utilisateur sélectionne une interface réseau, saisit un type de paquets ainsi qu'un intervalle de mesure. L'analyse de la distribution se traduit par la génération d'un tableau de résultats mais également d'une représentation graphique de la distribution telle que présentée à la figure 8.5.

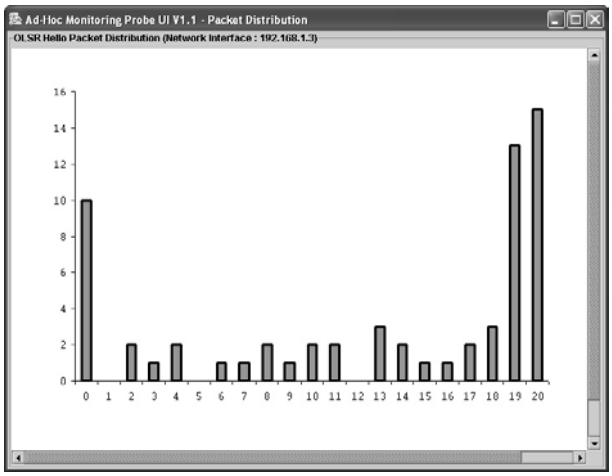


FIG. 8.5 – Distribution de paquets OLSR de type HELLO pour une interface donnée

Le chapitre 7 a décrit comment une analyse fine du plan de routage peut être exploitée dans le cadre de la gestion de fautes. Plus exactement, nous analysions la distribution des paquets HELLO dans le plan de routage OLSR afin de déceler des nœuds pathologiques. Un comportement pathologique se manifestait par une intermittence anormalement élevée du nœud pouvant être induit par de multiples facteurs comme par exemple une panne de routage, des problèmes de batterie, des perturbations physiques fortes.

Nous avons élaboré un scénario d'expérimentation dans lequel le fonctionnement d'un nœud ad-hoc a été volontairement dégradé en désactivant aléatoirement le démon de routage sur de courts intervalles de temps. La méthode de détection a été expérimentée en utilisant le module d'analyses statistiques afin de caractériser la distribution des paquets HELLO et d'identifier le nœud pathologique.

8.4.2 Participation au routage

Le module est également capable d'évaluer la participation au routage d'un nœud en s'appuyant sur la métrique que nous avions définie dans le cadre de la gestion de performance. D'une part, le module détermine si les nœuds ad-hoc assurent une retransmission correcte des paquets : lorsqu'un nœud ad-hoc reçoit un paquet à retransmettre, le module d'analyse vérifie si une retransmission a bien lieu *a posteriori*. D'autre part, une analyse du trafic émis par le nœud permet de déterminer la part du trafic générée par les retransmissions. Les mesures effectuées constituent une approximation de notre métrique. En effet, nous nous plaçons dans le contexte d'une sonde ad-hoc qui effectue une mesure indirecte des paquets retransmis par un nœud : le monitorage peut être biaisé par les perturbations physiques et les collisions. De façon similaire à l'approche *watchdog/pathrater* [134] , le nœud agent-sonde peut ne pas entendre une retransmission vers le nœud suivant à cause d'une collision alors que le nœud considéré a correctement

reçu le paquet.

8.4.3 Centralité des nœuds

Le module permet aussi d'effectuer une analyse de la centralité des nœuds au sein du réseau ad-hoc. L'utilisateur peut sélectionner deux formes de centralité : la centralité par degré et la centralité par vecteur propre [33]. Nous avons exploité ces méthodes d'analyse dans le cadre de l'organisation du plan de gestion et de la gestion de performance. La centralité par degré est une forme simple correspondant au degré du nœud dans un graphe. Un nœud est d'autant plus considéré comme central qu'il est connecté à un grand nombre de nœuds voisins. La centralité par

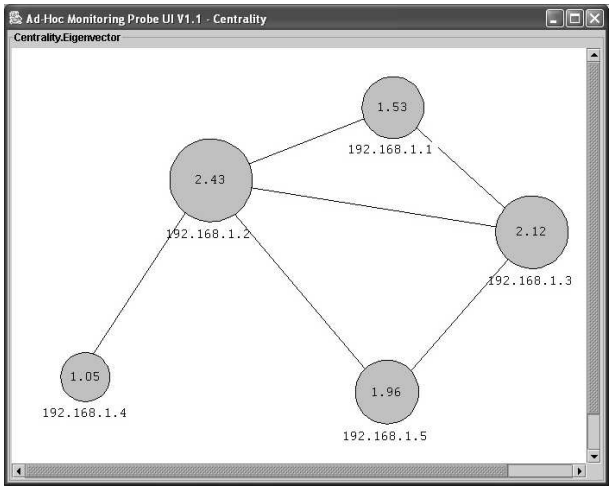


FIG. 8.6 – Analyse de la centralité des nœuds du réseau ad-hoc

vecteur propre est une forme plus élaborée définie de manière récursive : un nœud est d'autant plus central qu'il est lui-même connecté à des nœuds qui sont eux-mêmes centraux. L'algorithme a été implanté en utilisant la bibliothèque JAMA dédiée aux calculs d'algèbre linéaire [108]. Un exemple de résultats d'exécution est présenté à la figure 8.6. Les nœuds du réseau ad-hoc sont marqués à l'aide d'une valeur de centralité : par exemple, le nœud dont l'interface est 192.168.1.2 est celui disposant de la valeur de centralité la plus importante valant 2.43. Pour faciliter la lecture des résultats, la taille d'un nœud est définie proportionnellement à sa valeur.

8.5 Synthèse

Nos différents travaux de recherche ont été mis en œuvre dans le cadre du développement d'une plate-forme de monitorage, l'objectif étant de compléter nos travaux de simulations et d'analyses par une expérimentation pratique. Cette plate-forme dispose d'une structure composée d'un gestionnaire principal complété par un sous-ensemble d'agents-sondes implantés parmi les nœuds du réseau ad-hoc.

Le composant gestionnaire et le composant agent-sonde ont été prototypés afin d'évaluer la plate-forme de bout en bout : de la capture de paquets par l'agent-sonde à la vue synthétique générée par le gestionnaire via l'interface graphique utilisateur. L'essentiel du protypage a été réalisé à l'aide du langage de programmation Java, à l'exception de la capture de trames qui s'opère avec le langage C en s'appuyant sur la bibliothèque de fonctions *libpcap*.

Cette mise en œuvre nous a permis d'évaluer la généricité de notre modèle d'information à travers sa correcte instanciation sur un prototype. Par ailleurs, le développement d'un module d'analyses statistiques, en complément des fonctionnalités de base de la plate-forme, a permis d'expérimenter les différentes méthodes que nous avions définies pour la gestion des réseaux et services ad-hoc, telles que l'analyse de la centralité par vecteur propre, le filtrage des données de gestion et l'analyse de la distribution de paquets dans le plan de routage.

Les travaux présentés dans ce chapitre sont directement liés à nos publications [13, 19, 17].

Conclusion

Chapitre 9

Conclusion générale

9.1 Résumé des contributions

La supervision des réseaux ad-hoc représente une tâche définitivement plus problématique et complexe que celle des réseaux fixes traditionnels. En effet, la topologie du réseau est soumise à des changements très fréquents dans le temps. De plus, les stations mobiles qui le composent sont caractérisées par de multiples contraintes en termes d'énergie, de bande passante et de capacités système. Nous avons proposé dans cette thèse une nouvelle approche de supervision pour les réseaux et services ad-hoc capable de prendre en compte ces contraintes. Notre objectif était de répondre à un triple besoin d'intégration, de flexibilité et d'économie. Ainsi, notre approche permet d'intégrer les réseaux ad-hoc dans une démarche de gestion homogène, à travers la définition d'un modèle d'information générique. Elle permet également d'offrir une plus grande flexibilité à l'architecture et aux opérations de gestion, grâce à l'analyse des relations qui s'établissent entre les nœuds et à une adaptation dynamique aux changements du réseau. Enfin, elle permet de réduire les coûts de gestion en exploitant efficacement les informations fournies par le plan de routage.

Nos travaux de recherche comprennent trois axes principaux qui portent respectivement sur (1) la construction d'un modèle d'information générique pour les réseaux ad-hoc, (2) une réorganisation souple et à faible coût du plan de gestion et enfin (3) l'adaptation des opérations de gestion aux contraintes de ces réseaux.

9.1.1 Modélisation de l'information de gestion

Le premier axe de notre travail a consisté à définir un modèle d'information complet pour les réseaux ad-hoc. Les architectures de gestion ont jusqu'à présent laissé peu de place à la spécification des réseaux ad-hoc au sein du modèle d'information. Nous avons identifié les éléments caractéristiques d'un tel réseau et les avons décrits à l'aide d'un formalisme standard.

Nous avons défini pour ce faire une extension du modèle commun de l'information de gestion (CIM) pour prendre en charge les réseaux et services ad-hoc. Le modèle CIM définit un cadre formel pour décrire les ressources gérées et dispose d'un ensemble de schémas de référence dont les classes sont réutilisables et extensibles. Nous avons proposé un schéma d'extension qui peut être facilement instanciable sur différentes architectures de supervision. Nous y décrivons en particulier l'organisation du réseau ad-hoc, typiquement répartie sous forme de clusters, les échanges au sein du réseau à différentes échelles et la participation des nœuds au fonctionnement du réseau sur le plan de la consommation de ressources et sur celui de l'usage de services.

La standardisation récente des protocoles de routage ad-hoc nous a permis de compléter et de spécialiser notre modèle d'information. Nous avons en effet spécifié un sous-schéma d'extension pour le protocole de routage proactif OLSR. A l'instar des protocoles de routage tels que BGP et OSPF déjà intégrés au modèle commun, nous avons précisé les différentes propriétés du protocole incluant les informations relatives à la détection du voisinage, à la sélection des relais multipoints, à la diffusion des informations de topologie et à la maintenance des tables de routage. Nous avons aussi intégré les informations propres à la prise en charge d'interfaces multiples et d'interfaces non OLSR.

9.1.2 Réorganisation du plan de gestion

Le second axe de notre travail portait sur la réorganisation du plan de gestion en s'appuyant sur l'analyse des relations entre les nœuds. Le plan de gestion doit être suffisamment souple pour s'adapter dynamiquement aux changements de topologie. Il doit également permettre une répartition efficace de la charge de gestion sur les nœuds afin de minimiser son impact sur la performance du réseau.

Nous avons proposé une nouvelle organisation du plan de gestion reposant sur une méthode probabiliste. Vouloir superviser l'ensemble des nœuds du réseau est une contrainte trop forte et en inadéquation avec la nature même des réseaux ad-hoc. Nous avons donc relâché cette contrainte en limitant volontairement le nombre de nœuds ad-hoc gérés. Cette approche sélective consiste à identifier des composantes spatio-temporelles correspondant à des sous-ensembles de nœuds ayant une forte probabilité de voisinage dans le temps. Nous ne cherchons pas à ce que les nœuds soient connectés en permanence sur une période de temps, mais considérons une démarche plus souple où nous déterminons les nœuds qui ont été voisins durant un pourcentage de temps minimum. Cette méthode est qualifiée de *probabiliste* car son fonctionnement n'est garanti que de manière stochastique : nous garantissons avec une certaine probabilité qu'un pourcentage donné de nœuds sera géré.

Nous avons détaillé les algorithmes de gestion sous-jacents permettant l'extraction des composantes et l'élection de nœuds gestionnaires à travers différents mécanismes électifs fondés sur la notion de centralité : centralité de degré et centralité par vecteur propre. Ceux-ci permettent de sélectionner les nœuds ayant une importance structurelle dans le plan de gestion. En particulier, la centralité par vecteur propre prend en compte récursivement l'importance relative des nœuds : un nœud est d'autant plus important qu'il est connecté à des nœuds voisins eux-mêmes importants. En utilisant ces algorithmes, nous avons intégré notre démarche probabiliste au sein de l'architecture ANMP à travers un module de clusterisation additionnel. Ce module fonctionne en exploitant les informations du plan de routage fournies par le protocole OLSR.

9.1.3 Adaptation des opérations de gestion

Le dernier axe de notre travail s'intéressait à l'adaptation des opérations de gestion aux contraintes des réseaux ad-hoc. Dans cet objectif, nous avons utilisé différentes techniques relatives aux filtres, aux graphes et aux distributions statistiques pour évaluer l'état et les dépendances des nœuds. Nous avons essentiellement considéré les aires fonctionnelles correspondant à la gestion de performances et à la gestion de fautes.

Gestion de performances

Les propriétés et les tâches (notamment l'activité de routage) des nœuds ad-hoc évoluent dynamiquement dans le temps. Nous avons conçu deux méthodes d'analyse permettant de

construire une vue synthétique de l'état fonctionnel du réseau à partir des mesures de performance. La première méthode consiste à traiter les mesures à l'aide d'un filtrage de similitude ou de différence pour faire apparaître des patrons de trafic et de comportement, notamment identifier les *backbones* où l'activité de routage se concentre. La seconde méthode offre une analyse plus fine à base de graphes de dépendances afin de déterminer l'impact des nœuds sur le fonctionnement global du réseau en mettant en évidence les disparités entre nœuds. Cet impact peut être positif (les nœuds qui ont une activité de routage importante et qui font partie d'un *backbone*) ou au contraire négatif (les nœuds qui refusent d'intervenir comme routeurs ou qui consomment abusivement les ressources du réseau).

Gestion de fautes

Nous avons également introduit une méthode pour la gestion de fautes. Elle consiste à analyser l'intermittence des nœuds ad-hoc. Cette intermittence peut provenir de causes bénignes et inhérentes à ce type de réseau telles que la mobilité ou la dégradation temporaire de la connectivité. Elle est cependant pathologique lorsqu'elle est causée par des pannes de routage, des problèmes de batterie ou des perturbations physiques fortes. Nous souhaitions distinguer ces deux cas de figure afin de détecter par inférence les nœuds pathologiques. Nous avons défini une mesure de la théorie de l'information permettant localement d'identifier les nœuds intermittents dans le plan de routage. Un ensemble de méthodes collaboratives nous a permis de détecter les comportements pathologiques de manière distribuée au sein du réseau. L'approche proposée intègre un mécanisme d'auto-configuration utilisant l'algorithme de classification des *k-means* pour paramétrer dynamiquement la méthode de détection.

9.1.4 Expérimentations de nos travaux

Indépendamment des modèles analytiques mis en œuvre, nos différentes propositions de gestion ont été évaluées à travers un ensemble de simulations et le prototypage d'une plateforme.

Evaluation par la simulation

Nous avons évalué les performances de la réorganisation du plan de supervision et de l'adaptation des opérations de gestion. Il s'agissait notamment de déterminer la part de nœuds couverts avec la méthode probabiliste et de mesurer la sensibilité et la spécificité de la détection de fautes. De nombreuses séries d'expériences ont été réalisées à l'aide du simulateur réseau ns-2 [74] qui est largement répandu dans la communauté[4], même s'il requiert certaines précautions d'usage. Les méthodes de gestion ont pu être expérimentées avec différents modèles de mobilité, en l'occurrence le modèle de mobilité RWP (*Random WayPoint*) *steady-state* et le modèle de mobilité de groupe RPGM (*Reference Point Group Mobility*) ainsi qu'à travers différents scénarios : établissement d'un *backbone*, connexion à une passerelle Internet, nœuds non collaboratifs, nœuds intermittents, comportement de groupes et dégradation de performances.

[4]Une étude statistique [171] sur la base des publications de la conférence internationale ACM MobiHoc montre que ns-2 est le simulateur le plus utilisé avec 44,4% des cas, suivi par les simulateurs *faits maison* représentant 24,5% des cas.

Mise en œuvre d'un prototype

Un travail complémentaire a consisté à développer une plate-forme de monitorage. Nous avons prototypé un composant agent-sonde capable de capturer les paquets circulant sur le réseau et un composant gestionnaire capable de construire une vue synthétique du réseau ad-hoc. Nous avons également développé un module d'analyses statistiques offrant différentes fonctionnalités telles que l'analyse de la centralité par vecteur propre, le filtrage des données de gestion et l'analyse de la distribution de paquets dans le plan de routage. Cette mise en œuvre a permis d'évaluer notre modèle d'information à travers sa correcte instanciation et d'expérimenter les différentes techniques d'analyses utilisées dans le cadre de nos méthodes de gestion.

9.2 Perspectives

Les travaux de recherche réalisés au cours de cette thèse ouvrent différentes perspectives scientifiques à moyen et long termes. Ils contribuent à une meilleure prise en charge des réseaux ad-hoc dans le plan de gestion. Ils ouvrent également la voie, de manière plus générale, à de nouvelles optimisations pour la supervision des réseaux et services dynamiques.

9.2.1 Construire un cadre théorique complet

Cette thèse participe tout d'abord à la définition d'un cadre théorique pour la gestion des réseaux et services ad-hoc. Nous avons en effet proposé différents modèles analytiques au cours de nos travaux de recherche pour l'organisation du plan de gestion et l'adaptation des opérations. Les premiers travaux théoriques en la matière ont été réalisés par Mark Burgess [40] pour évaluer le passage à l'échelle de différents modèles organisationnels dans le cadre de la gestion par politique. Une première perspective de recherche vise à construire un cadre théorique homogène et complet pour la supervision des réseaux ad-hoc. Ce cadre est indispensable pour quantifier les performances des méthodes de gestion en tenant compte des propriétés des réseaux ad-hoc. Il permettra de comparer la complexité et les facteurs d'échelle de ces méthodes en s'appuyant sur une même base de référence. Une seconde perspective porte plus particulièrement sur le raffinement des modèles utilisés dans le cadre de la gestion de fautes. Nous avons utilisé une modélisation à base de chaînes de Markov pour caractériser la perception d'un nœud pathologique par un nœud observateur. Cette modélisation peut être raffinée en utilisant une structure markovienne d'ordre supérieur [37]. La méthode de détection pourra également être évaluée avec de nouveaux modèles de mobilité, et une analyse formelle complémentaire permettra de déterminer l'impact direct des paramètres de mobilité sur la mesure d'entropie.

9.2.2 Etendre le modèle d'information aux autres protocoles de routage

Nous avons par ailleurs étendu le modèle commun de l'information CIM pour qu'il prenne en charge le protocole de routage OLSR et avons défini en conséquence une base d'informations de gestion OLSR-MIB. Cette spécification peut naturellement être étendue aux autres protocoles de routage ad-hoc qui ont été récemment standardisés par le groupe de travail MANET. Notre objectif vise plus particulièrement à intégrer un protocole réactif tel que le protocole AODV. Dans le cas d'un protocole proactif, chaque nœud du réseau maintient une table de routage et les mises à jour de la table sont obtenues par échange périodique d'informations de topologie entre les nœuds. Aussi, nous avons décrit pour le protocole OLSR les différents mécanismes sous-jacents, notamment la sélection des relais multipoints et la maintenance des tables de routage. Dans le

cas d'un protocole réactif, les nœuds ad-hoc ne maintiennent pratiquement pas d'informations sur la topologie du réseau et les routes sont établies à la demande des nœuds par diffusion d'une requête. Une perspective consistera à intégrer le protocole de routage AODV dans le modèle commun de l'information en décrivant ce mécanisme de découverte, et en standardisant les bases d'informations de gestion associées.

9.2.3 Introduire de nouveaux critères au sein du modèle organisationnel

Le modèle organisationnel peut également être perfectionné en s'appuyant sur une sélection multi-critères [177]. Nous avons réorganisé le plan de gestion en utilisant une méthode probabiliste qui permet de relâcher les contraintes sur le plan de gestion en fonction des propriétés spatiales et temporelles des nœuds du réseau. Une perspective de ce travail consiste à raffiner l'approche probabiliste dans le cadre plus classique d'une sélection multi-critères. Elle vise à prendre en compte des propriétés supplémentaires, comme par exemple le niveau d'énergie des batteries et la puissance de calcul des nœuds, pour la sélection des gestionnaires et des agents. En particulier, elle permettra de minimiser l'implication des équipements de faible capacité (capteurs par exemple) dans la tâche de gestion. Par ailleurs, une autre perspective porte sur l'implantation de mécanismes de réputation [25] dans le plan de gestion. Ces mécanismes permettent typiquement d'évaluer la fiabilité d'un nœud à fournir un service. Dans notre contexte, ils permettront de déterminer la fiabilité d'un nœud à interagir en tant que gestionnaire et/ou agent, et à en prendre compte lors de l'affectation des rôles. Ces mécanismes peuvent aussi être mis en œuvre dans le cadre de l'adaptation des opérations de gestion. L'objectif consiste alors à améliorer les mesures de performances et la détection de fautes. En particulier, les méthodes peuvent être étendues en pondérant les mesures effectuées par les nœuds ad-hoc en fonction de leur niveau de fiabilité. Les mesures provenant de nœuds fiables auront des poids plus élevés et pourront donc être prises davantage en compte lors du processus de gestion.

9.2.4 Poursuivre une meilleure maîtrise du degré de liberté des nœuds

Enfin, les contributions de cette thèse peuvent s'inscrire dans un contexte plus large que celui des réseaux ad-hoc : les problèmes posés par ceux-ci constituent en réalité une projection des problèmes que nous pourrons rencontrer plus généralement avec les réseaux et services dynamiques de demain [84]. Un problème majeur porte sur le degré de liberté des nœuds du réseau. Dans les infrastructures fixes traditionnelles, les nœuds sont fortement contraints du fait de l'absence de mobilité et de leur fonction dédiée : les terminaux et les routeurs correspondent à des équipements spécifiques. Les réseaux de nouvelles générations offrent aux nœuds une plus grande liberté de mouvement grâce à la mobilité, ainsi qu'une plus grande liberté de fonction : les stations jouent à la fois les rôles de terminaux et de routeurs. Nous avons contribué à une meilleure maîtrise de ce degré de liberté, par une organisation plus souple du plan de gestion et par des méthodes de gestion capables d'évaluer le comportement des nœuds au sein du réseau. Cette refonte est essentielle pour offrir une plus grande flexibilité au plan de gestion. En particulier, nous pourrions poursuivre cette démarche en définissant un modèle de gestion *fish-eye* [155] où la nature et la fréquence des opérations de gestion opérées par le gestionnaire auprès d'un agent seraient automatiquement définies en fonction de la distance qui le sépare de celui-ci. Typiquement, les opérations de configuration seraient limitées au voisinage proche tandis que les opérations de monitorage pourraient s'opérer sur une population de nœuds plus large.

9.3 Publications relatives

Journaux internationaux

- Badonnel R., State R., Festor O. and Schaff A., A Framework for Optimizing the End-to-End Connectivity in Mobile Ad-Hoc Networks, *Journal of Network and Systems Management (JNSM)*, Volume 13, Number 4, Springer Verlag / Plenum Publishing Corporation, Décembre 2005
- Badonnel R., State R. and Festor O., Management of Mobile Ad-Hoc Networks : Information Model and Probe-based Architecture, *ACM International Journal of Network Management (ACM IJNM)*, Volume 15, Number 5, Septembre 2005
- Badonnel R., State R. and Festor O., Self-Organized Monitoring in Ad-Hoc Networks, *Telecommunication Systems*, Volume 30, Number 1, Springer Verlag, Novembre 2005

Conférences internationales avec actes et comité de lecture

- Badonnel R., State R., Festor O., Probabilistic Management of Ad-Hoc Networks, *in Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS'06)*, Vancouver, Canada, Avril 2006, ***Best Student Paper Award***
- Badonnel R., State R., Festor O., Fault Monitoring in Ad-Hoc Networks Based on Information Theory, *in Proceedings of the 5th International IFIP-TC6 Networking Conference (NETWORKING'06)*, Lecture Notes in Computer Science 3976, Coimbra, Portugal, Mai 2006, ***taux d'acceptation : 20.4%***
- Badonnel R., State R., Festor O., Management of Mobile Ad-Hoc Networks : Evaluating the Network Behavior, *in Proceedings of the 9th IEEE/IFIP International Symposium on Integrated Network Management (IM'05)*, Nice, France, Mai 2005, ***taux d'acceptation : 23.5%***
- Badonnel R., State R., Festor O., Monitoring End-to-End Connectivity in Mobile Ad-Hoc Networks, *in Proceedings of the 4th International Conference on Networking*, Lecture Notes in Computer Science 3421, Springer Verlag, France, Avril 2005
- Keller A., Badonnel R., Automating the Provisioning of Application Services with the BPEL4WS Workflow Language, *in Proceedings of the 15th IFIP/IEEE International Workshop on Distributed Systems : Operations and Management (DSOM'04)*, Lecture Notes in Computer Science 3278, Springer Verlag, Davis, CA, USA, Novembre 2004, ***taux d'acceptation : 22.2%***

Conférences nationales avec actes et comité de lecture

- Badonnel R., State R., Festor O., Schaff A., Evaluer l'Impact d'un Nœud au sein d'un Reseau Ad-Hoc, *in Proceedings of the Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'05)*, Bordeaux, France, Mars 2005
- R.Badonnel, L. Andrey and O. Festor, Composition de Services et Supervision. *in Proceedings of Les Nouvelles Technologies de la Répartition (NOTERE'04)*, Juillet 2004, Saïdia, Maroc

Séminaires

- Badonnel R., State R., Festor O., Monitorage de Fautes dans les Reseaux Ad-Hoc, Journées Pôle Réseaux et Communications, LIFL-CNRS, Lille, France, Mars 2006
- Badonnel R., State R., Festor O., Composition de Services et Supervision : Application aux Services Web, Journées Réseau Grand Est, LORIA-INRIA Lorraine, Nancy, France, Octobre 2004

Travaux en cours de soumission

- Badonnel R., State R., Festor O., Management of Ad-Hoc Networks, *INRIA Research Report*, Février 2006, chapitre de livre en cours de soumission pour l'ouvrage international Elsevier Handbook on Network Management
- Badonnel R., State R., Festor O., A Probabilistic Approach for Managing Mobile Ad-Hoc Networks, *INRIA Research Report*, Avril 2006, papier invité à être soumis au journal international IEEE eTransactions on Network and Service Management
- Badonnel R., State R., Festor O., A Self-Configuring Fault Monitoring Scheme for Ad-Hoc Networks, *INRIA Research Report*, Mai 2006, papier invité à être soumis au journal international Elsevier Ad Hoc Networks

Brevet industriel relatif à la gestion en général

- Keller A., Badonnel R., Systems and Methods for Automated Concurrent Provisioning of Managed Resources, IBM Research Patent Disclosure filed with the USPTO, IBM Docket No. YOR920050066US1, Septembre 2005

Bibliographie

[1] Defense Advanced Research Project Agency (DARPA). http ://www.darpa.mil.

[2] Distributed Management Task Force (DMTF). http ://www.dmtf.org.

[3] Internet Engineering Task Force (IETF). http ://www.ietf.org.

[4] IP Performance Metric (IPPM) Working Group, Internet Engineering Task Force. http ://www.ietf.org/html.charters/ippm-charter.html.

[5] OLSR Extension for ns-2. Navy Research Laboratory, Networks and Communication Systems Branch, Washington, DC, USA, http ://pf.itd.nrl.navy.mil/projects/olsr/.

[6] TCPDUMP/LIBPCAP. Lawrence Berkeley National Laboratory, Berkeley, CA, USA, http ://www.tcpdump.org.

[7] M. Aida, N. Miyoshi, and K. Ishibashi. A Scalable and Lightweight QoS Monitoring Technique Combining Passive and Active Approaches. In *Proc. of the 22nd IEEE International Conference on Computer Communications (INFOCOM'03)*, San Francisco, CA, USA, April 2003.

[8] R. V. Hogg andE. A. Tanis. *Probability and Statistics for Engineers and Scientists*. Prentice Hall ; 8th edition, February 2006.

[9] G. Armitage. *Quality of Service in IP Networks*. Sams Publishing, April 2000.

[10] G. Armitage. *Wireless Network Performance Handbook*. McGraw-Hill Network Engineering, May 2003.

[11] A.Veres, A.T Campbell, and M.Barry. Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control. *IEEE Journal on Selected Areas in Communications*, 19(10), October 2001.

[12] R. Badonnel, R. State, and O. Festor. Management of Mobile Ad-Hoc Networks : Evaluating the Network Behavior. In *Proc. of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM'05)*, pages 17–30, Nice, France, April 2005. IEEE Communications Society. Acceptance Rate : 23.5%.

[13] R. Badonnel, R. State, and O. Festor. Management of Mobile Ad-Hoc Networks : Information Model and Probe-based Architecture. *ACM International Journal of Network Management (ACM IJNM)*, 15(5), September 2005.

[14] R. Badonnel, R. State, and O. Festor. Monitoring End-to-End Connectivity in Mobile Adhoc Networks. In *Proc. of the 4th IEEE International Conference on Networking*, France, April 2005. Lecture Notes in Computer Science 3421, Springer Verlag.

[15] R. Badonnel, R. State, and O. Festor. A Probabilistic Approach for Managing Mobile Ad-Hoc Networks. Technical report, LORIA/INRIA, April 2006. papier invité à être soumis au journal international IEEE eTransactions on Network and Service Management.

[16] R. Badonnel, R. State, and O. Festor. A Self-Configuring Fault Monitoring Scheme for Ad-Hoc Networks. Technical report, LORIA/INRIA, May 2006. papier invité à être soumis au journal international Elsevier Ad Hoc Networks.

[17] R. Badonnel, R. State, and O. Festor. Fault Monitoring in Ad-Hoc Networks Based on Information Theory. In *Proc. of the 5th International IFIP-TC6 Networking Conference (NETWORKING'06)*, Coimbra, Portugal, May 2006. Lecture Notes in Computer Science 3976, Springer Verlag. Acceptance Rate : 20.4%.

[18] R. Badonnel, R. State, and O. Festor. Management of Ad-Hoc Networks. Technical report, LORIA/INRIA, February 2006. chapitre de livre en cours de soumission pour l'ouvrage international Elsevier Handbook on Network Management.

[19] R. Badonnel, R. State, and O. Festor. Probabilistic Management of Ad-Hoc Networks. In *Proc. of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS'06)*, Vancouver, Canada, April 2006. Best Student Paper Award.

[20] R. Badonnel, R. State, O. Festor, and A. Schaff. Evaluer l'Impact d'un Nœud au sein d'un Reseau Ad-Hoc. In *Proc. of the Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'05)*, Bordeaux, France, March 2005.

[21] R. Badonnel, R. State, O. Festor O, and A. Schaff. A Framework for Optimizing the End-to-End Connectivity in Mobile Ad-Hoc Networks. *Journal of Network and Systems Management (JNSM)*, 13(4), December 2005.

[22] F. Bai, N. Sadagopan, and A. Helmy. IMPORTANT : a Framework to Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for Ad-Hoc Networks. In *Proc. of the 22nd IEEE International Conference on Computer Communications (INFOCOM'03)*, San Francisco, CA, USA, April 2003.

[23] L. Bao and J. J. Garcia-Luna-Aceves. Topology Management in Ad-Hoc Networks. In *Proc. of the 4th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MOBIHOC'03)*, pages 129–140, Annapolis, MD, USA, June 2003.

[24] W. Barth. *Nagios : System and Network Monitoring*. NS Press, U.S. Edition, May 2006.

[25] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, editors. *Mobile Ad Hoc Networking*. IEEE Press and John Wiley & Sons, Inc., Piscataway, NJ and New York, NY, April 2004.

[26] P.A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addison Wesley, 1987.

[27] C. Bettstetter, G. Resta, and P. Santi. Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks. *IEEE Transactions On Mobile Computing*, 2(3), July 2003.

[28] Christian Bettstetter, Hannes Hartenstein, and Xavier Perez-Costa. Stochastic Properties of the Random Waypoint Mobility Model. *ACM/Kluwer Wireless Networks, Special Issue on Modeling and Analysis of Mobile Networks*, 2003.

[29] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal of Selected Areas in Communications*, 18(3), March 2005.

[30] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. http ://www.ietf.org/rfc/rfc2475.txt, December 1998. IETF Request for Comments 2475.

[31] John Bloomer. *Power Programming with RPC (Nutshell Handbooks)*. O'Reilly, 1992.

[32] P. Bonacich. Factoring and Weighing Approaches to Status Scores and Clique Identification. *Journal of Mathematical Sociology*, 2 :113–120, 1972.

[33] P. Bonacich and P. Lloyd. Eigenvector-like Measures of Centrality for Asymmetric Relations. *Social Networks*, 23 :191–201, 2001.

[34] S.P. Borgatti and M.G. Everett. A Graph-theoric Perspective on Centrality. *Social Networks*, 2006.

[35] J. Y. Le Boudec. On the Stationary Distribution of Speed and Location of Random Waypoint. *IEEE Transactions on Mobile Computing*, 4(4), 2005.

[36] J.-Y. Le Boudec and M. Vojnovic. Perfect Simulation and Stationarity of a Class of Mobility Models. In *Proc. of 24th IEEE International Conference on Computer Communications (INFOCOM'05)*, Miami, FL, USA, March 2005.

[37] P. Bremaud. *Markov Chains*. Springer Publishing, January 2001.

[38] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proc. of The 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, Dallas, TX, USA, October 1998.

[39] W. Bumpus, J. W. Sweitzer, P. Thompson, A. R. Westerinen, and R. C. Williams. *Common Information Model*. John Wiley and Sons, 2000.

[40] M. Burgess and G. Canright. Scalability of Peer Configuration Management in Logically Ad-hoc Networks. *E-Transactions on Network and Service Management (eTNSM)*, 1(1), April 2004.

[41] M. Burmester and A. Yasinsac. *Security Issues in Ad-Hoc Networks*. Springer Verlag, August 2006.

[42] L. Buttyan and J. P. Hubaux. Enforcing Service Availability in Mobile Ad-Hoc WANs. In *Proc. of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MOBIHOC'00)*, Boston, MA, USA, August 2000.

[43] L. Buttyan and J. P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *ACM/Kluwer Mobile Networks and Applications*, 8(5), October 2003.

[44] J. Byers and G. Nasser. Utility-Based Decision-Making in Wireless Sensor Networks,. Technical report, Computer Science Department, Boston University, Boston, MA, USA, June 2000.

[45] M. Cagalj, S. Ganeriwal, I. Aad, and J.P. Hubaux. On Selfish Behavior in CSMA/CA Networks. In *Proc. of the 24th IEEE International Conference on Computer Communications (INFOCOM'05)*, Miami, FL, USA, March 2005.

[46] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Researchl. *Wireless Communications and Mobile Computing, Special Issue on Mobile Ad Hoc Networking : Research, Trends and Applications*, 2(5), 2002.

[47] T. Camp, W. Navidi, and N. Bauer. Improving the Accuracy of Random Waypoint Simulations through Steady-state Initialization. In *Proc. of the 15th International Conference on Modeling and Simulation (MS'04)*, Montreal, Quebec, Canada, March 2004.

[48] J. D. Case, M. Fedor, M. L. Schoffstall, and J. Davin. Simple Network Management Protocol (SNMP). http ://www.ietf.org/rfc/rfc1157.txt, May 1990. IETF Request for Comments 1157.

[49] R. Chadha and H. Cheng. Policy-Based Mobile Ad Hoc Network Management for DRAMA. In *Proc. of IEEE Military Communications Conference (MILCOM'04)*, Monterey, CA, USA, October 2004.

[50] R. Chadha, H. Cheng, Yuu-Heng Chend, and J. Chiang. Policy-based Mobile Ad-hoc Network Management. In *Proc. of the 5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'04)*, New York, USA, June 2004.

[51] M. Chaffangeon, S. Reichert, and P. Y. Thomas. Conception d'une Architecture de Monitorage pour les Réseaux Sans-Fil Ad-Hoc. Technical report, Ecole Supérieure d'Informatique et Applications de Lorraine (ESIAL), Nancy, France, 2004.

[52] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, and A. Smith. COPS Usage for Policy Provisioning (COPS-PR). http ://www.ietf.org/rfc/rfc3084.txt, March 2001. IETF Request for Comments 3084.

[53] W. Chen, N. Jain, and S. Singh. ANMP : Ad-Hoc Network Management Protocol. *IEEE Journal on Selected Areas in Communications (JSAC)*, 17(8) :1506–1531, August 1999.

[54] S. Chien-Chung, C. Srisathapornphat, L. Rui, H. Zhuochuan, C. Jaikaeo, and E. Lloyd. CLTC : a Cluster-Based Topology Control for Ad-Hoc Networks. *IEEE Transactions on Mobile Computing*, 3(1) :18–32, January 2004.

[55] D. Minh Chiu, R. Sudama, and D. M. Chiu. *Network Monitoring Explained : Design and Application.* Ellis Horwood Series in Computer Communications and Networking, Prentice Hall PTR, July 1992.

[56] Common Information Model (CIM) Version 2.9. Distributed Management Task Force (DMTF), http ://www.dmtf.org/standards/cim, January 2005.

[57] T. Clausen and P. Jacquet. Optimized Link State Routing (OLSR) Protocol. http ://www.ietf.org/rfc/rfc3626.txt, October 2003. IETF Request for Comments 3626.

[58] A. Clemm and A. Bansal. Auto-Discovery at the Network and Service Management Layer. In *Proc. of the 8th IFIP/IEEE International Symposium on Integrated Network Management (IM'03)*, pages 365 – 378, Colorado Springs, Colorado, USA, March 2003. IEEE Communications Society.

[59] G. Combs. Ethereal : A Network Protocol Analyzer. http ://www.ethereal.com.

[60] The ASN.1 Consortium. Abstract Syntax Notation One (ASN.1). http ://www.asn1.org/.

[61] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* Second Edition, MIT Press and McGraw-Hill, 2001.

[62] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley & Sons, 1991.

[63] D. Cvetkovic, P. Rowlinson, and S. Simic. *Eigenspaces of Graphs (Encyclopedia of Mathematics and its Applications).* Cambridge University Press, November 2004.

[64] Réseau National de Recherche en Télécommunications. Projet SAFARI : Services Ad hoc/Filaires : Développement d'une Architecture de Réseau intégré. http ://www.rnrt.org/rnrt/projets/.

[65] A. Dennis, B. H. Wixom, and D. Tegarden. *Systems Analysis and Design with UML Version 2.0 : An Object-Oriented Approach.* John Wiley and Sons, 2005.

[66] A. Deshpande, C. Guestrin, and S. Madden. Using Probabilistic Models for Data Management in Acquisitional Environments. In *Proc. of the Second Biennial Conference on Innovative Data Systems Research)*, Asilomar, CA, USA, January 2005.

[67] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-based Approximate Querying in Sensor Networks. *VLDB Journal*, 14(4), November 2005.

[68] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Model-Driven Data Acquisition in Sensor Networks. In *Proc. of the 13th International Conference on Very Large Data Bases (VLDB'04)*, pages 588–599, August 2004.

[69] R. Diestel. *Graph Theory*. Springer ; 2nd edition, February 2000.

[70] R. Droms and T. Lemon. *The DHCP Handbook : Understanding, Deploying, and Managing Automated Configuration Services*. MacMillan Publishing Company, October 1999.

[71] R. D'Souza, S. Ramanathan, and D. Temple Land. Measuring Performance of Ad-hoc Networks using Timescales for Information Flow. In *Proc. of the 22nd IEEE International Conference on Computer Communications (INFOCOM'03)*, San Francisco, CA, USA, April 2003.

[72] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. The COPS (Common Open Policy Service) Protocol. http ://www.ietf.org/rfc/rfc2748.txt, April 1999. IETF Request for Comments 2748.

[73] R. Enns. NETCONF Configuration Protocol. http ://www.ietf.org/internet-drafts/draft-ietf-netconf-prot-12.txt, February 2006. IETF Internet Draft.

[74] K. Fall and K. Varadhan. The Network Simulator (NS) Manual : Formerly Known as NS Notes and Documentation. http ://www.isi.edu/nsnam/ns/.

[75] D. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli. A Proposed Standard for Role Based Access Control. *ACM Transactions on Information and System Security*, 4(3), August 2001.

[76] O. Festor and A. Schaff. *Standards pour la Gestion des Réseaux et des Services*. IC2 Réseaux et Télécoms, Hermès Science Publishing, January 2004.

[77] D. Flanagan. *Java In A Nutshell*. O'Reilly Media, March 2005.

[78] International Organisation for Standardization. Information Processing Systems, Open Systems Interconnection, Basic Reference Model - Part 4 : Management Framework. ISO IEC 7498-4, November 1989.

[79] Distributed Management Task Force. Common Information Model (CIM) Core Model White Paper version 2.4. DSP0111, August 2000.

[80] Distributed Management Task Force. Common Information Model (CIM) Application White Paper version 2.7. DSP0140, June 2003.

[81] Distributed Management Task Force. Common Information Model (CIM) Concepts White Paper version 2.4+. DSP0110, June 2003.

[82] Distributed Management Task Force. Common Information Model (CIM) Network White Paper version 2.7. DSP0152, June 2003.

[83] Distributed Management Task Force. Common Information Model (CIM) System White Paper version 2.7. DSP0150, June 2003.

[84] B. Ford. Unmanaged Internet Protocol : Taming the Edge Network Management Crisis. In *Proc. of the 2nd ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets-II)*, Cambridge, MA, USA, November 2003.

[85] L.C. Freeman. Centrality in Networks : Conceptual Clarification. *Journal of Social Networks*, 1, 1979.

[86] N. E. Friedkin. Theoretical Foundations for Centrality Measures. *American Journal of Sociology*, 96, 1991.

[87] A. El Gamal, J. P. Mammen, B. Prabhakar, and D. Shah. Throughput-Delay Trade-off in Wireless Networks. In *Proc. of the 23rd IEEE International Conference on Computer Communications (INFOCOM'04)*, Hong Kong, March 2004.

[88] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns : Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, January 1995.

[89] Y. Ganjali and A. Keshavarzian. Load Balancing in Ad Hoc Networks : Single-path Routing vs. Multi-path Routing. In *Proc. of the 23rd IEEE International Conference on Computer Communications (INFOCOM'04)*, Hong Kong, March 2004.

[90] M. S. Gast. *802.11 Wireless Networks : The Definitive Guide, Chapter 3 : 802.11 MAC Fundamentals*. O'Reilly Media, , Second Edition, April 2005.

[91] M. S. Gast. *802.11 Wireless Networks : The Definitive Guide, Chapter 9 : Contention-Free Service with the PCF*. O'Reilly Media, , Second Edition, April 2005.

[92] M. S. Gast. *802.11 Wireless Networks : The Definitive Guide, Second Edition*. O'Reilly Media, April 2005.

[93] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, July 1985.

[94] V. C. Giruka and M. Singhal. Hello Protocols for Ad-Hoc Networks : Overhead and Accuracy Tradeoffs. In *Proc. of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)*, Taormina, Italy, jun 2005.

[95] Savo G. Glisic. *Advanced Wireless Networks : 4G Technologies*. John Wiley and Sons, Hoboken, NJ, USA, 2006.

[96] G. Goldszmidt and Y. Yemini. Delegated Agents for Network Management. *IEEE Transaction on Communications*, 36, March 1998.

[97] S. Gouveris, S. Sivavakeesar, G. Pavlou, and A. Malatras. Programmable Middleware for the Dynamic Deployment of Services and Protocols in Ad-Hoc networks. In *Proc. of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM'05)*, pages 3–16, Nice, France, April 2005. IEEE Communications Society.

[98] Global Positioning System. U.S. Department of Transportation and Department of Defense, http ://www.navcen.uscg.gov.

[99] Z.J. Haas. A New Routing Protocol for the Reconfigurable Wireless Networks. In *Proc. of the 6th IEEE International Conference on Universal Personal Communications (ICUPC '97)*, San Diego, CA, USA, October 1997.

[100] Z.J. Haas and J. Deng. On Optimizing the Backoff Interval for Random Access Schemes. *IEEE Transactions on Communications*, 51(12), December 2003.

[101] T. Hara. Effective Replica Allocation in Ad-Hoc Networks for Improving Data Accessibility. In *Proc. of the 20th IEEE International Conference on Computer Communications (INFOCOM'01)*, Anchorage, Alaska, USA, April 2001.

[102] M. Hasan, B. Sugla, and R. Viswanathan. A Conceptual Framework for Network Management Event Correlation and Filtering Systems. In *Proc. of the 6th IEEE/IFIP International Conference on Integrated Network Management (IM'99)*, Boston, MA, USA, May 1999.

[103] A. M. Jonassen Hass. *Configuration Management Principles and Practice*. Addison-Wesley Professional, December 2002.

[104] H.G. Hegering, S. Abeck, and B. Neumair. *Integrated Management of Networked Systems : Concepts, Architectures, and Their Operational Application*. Morgan Kaufmann Publisher, 1999.

[105] G. Held. *Wireless Mesh Networks*. Auerbach Publications, June 2005.

[106] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury. *Feedback Control of Computing Systems*. John Wiley and Sons, 2004.

[107] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba. Advanced Configuration and Power Interface (ACPI) Specification 3.0. http ://www.acpi.info/, December 2005.

[108] J. Hicklin, C. Moler, P. Webb, R. F. Boisvert, B. Miller, R. Pozo, and K. Remington. JAMA : Java Matrix Package. http ://math.nist.gov/javanumerics/jama.

[109] R. V. Hogg and E. A. Tanis. *Probability and Statistical Inference*. Prentice Hall ; 7th edition, January 2005.

[110] X. Hong, M. Gerla, G. Pei, and C. C. Chiang. A Group Mobility Model for Ad Hoc Wireless Networks. In *Proc. of the ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'99)*, Seattle, WA, USA, August 1999.

[111] C. Hunt. *TCP/IP Network Administration (3rd Edition ; O'Reilly Networking)*. O'Reilly Media, 3rd Edition, April 2002.

[112] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance of MultiPoint Relaying in Ad-Hoc Mobile Routing Protocols. In *Proc. of the IFIP International Networking Conference (Networking'05)*, Pisa, Italy, May 2002.

[113] P. Jacquet and W. Szpankowski. Entropy Calculation via Analytic Depoissonization. *IEEE Transaction on Information Theory*, 45 :1072–1081, 1999.

[114] C. Jelger and T. Noel. Proactive Address Autoconfiguration and Prefix Continuity in IPv6 Hybrid Ad Hoc Networks. In *Proc. of the 2nd IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON'05)*, Santa Clara, CA, USA, September 2005.

[115] M. Kahani and H. W. Peter Beadle. Decentralized Approaches for Network Management. *SIGCOMM Computer Communication Review*, 27(3), 1997.

[116] P. Kalyanasundaram, A. S. Sethi, C. M. Sherwin, and D. Zhu. A Spreadsheet-Based Scripting Environment for SNMP. In *Proc. of the fifth IFIP/IEEE International Symposium on Integrated Network Management (IM'97)*, London, UK, may 1997.

[117] D. S. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proc. of the 12th international conference on World Wide Web (WWW '03)*, Budapest, HUNGARY, May 2003.

[118] A. Keller and R. Badonnel. Automating the Provisioning of Application Services with the BPEL4WS Workflow Language. In *Proc. of the 15th IFIP/IEEE International Workshop on Distributed Systems : Operations and Management (DSOM'04)*, Davis, CA, USA, March 2005. Lecture Notes in Computer Science 3278, taux d'acceptation : 22.2%.

[119] A. Keller and R. Badonnel. Systems and Methods for Automated Concurrent Provisioning of Managed Resources. IBM Research Patent filed with the USPTO, IBM Docket No. YOR920050066US1, September 2005.

[120] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. *IEEE Computer*, 36(1), January 2003.

[121] B. W. Kernighan, D. Ritchie, and D. M. Ritchie. *The C Programming Language*. Prentice Hall PTR, March 1988.

[122] A. Kherani, E. Altman, P. Michiardi, and R. Molva. Non-cooperative Forwarding in Ad-hoc Networks. In *Proc. of the International IFIP Networking Conference (Networking'05)*, Waterloo, Canada, May 2005.

[123] D. R. Kosiur. *Understanding Policy-based Networking*. John Wiley and Sons, February 2001.

[124] H. Kreger, W.K. Harold, and L. Williamson. *Java and JMX : Building Manageable Systems,*. Adisson-Wesley Publishers, December 2002.

[125] P. Kyasanur and N. Vaidya. Detection and Handling of MAC Layer Misbehavior in Wireless Networks. Technical report, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, IL, USA, August 2002.

[126] B. Landfeldt, P. Sookavatana, and A. Seneviratne. The Case for a Hybrid Passive/Active network Monitoring Scheme in the Wireless Internet. In *Proc. of the 8th IEEE International Conference on Networks (ICON'00)*, Singapore, September 2000.

[127] A. Laouiti, A. Qayyum, and L. Viennot. Multipoint Relaying : An Efficient Technique for Flooding in Mobile Wireless Networks. In *Proc. of the 35th Annual Hawaii International Conference on System Sciences (HICSS'2002)*, Big Island, HI, USA, January 2002.

[128] H. Li, M. Lott, M. Weckerle, W. Zirwas, and E. Schulz. Multihop Communications in Future Mobile Radio Networks. In *Proc. of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'02)*, Lisboa, Portugal, September 2002.

[129] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. In *Proc. of the Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, CA, 1967.

[130] S. Mahaman and J.C. Tranquillin. Module d'Analyses Avancées pour le Monitorage des Réseaux Ad-Hoc. Technical report, Ecole Supérieure d'Informatique et Applications de Lorraine (ESIAL), Nancy, France, 2005.

[131] M. Maleki, K. Dantu, and M. Pedram. Lifetime Prediction Routing in Mobile Ad Hoc Networks. In *Proc. of the IEEE Wireless Communications and Networking Conf.(WCNC'03)*, New Orleans, LA, USA, March 2003.

[132] Mobile Ad-Hoc Networks (MANET) Working Group, Internet Engineering Task Force. http ://www.ietf.org/html.charters/manet-charter.html.

[133] MANET Autoconf Working Group, Internet Engineering Task Force. http ://www.ietf.org/html.charters/autoconf-charter.html.

[134] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating Routing Misbehavior in Mobile Ad-Hoc Networks. In *Proc. of the IEEE/ACM International Conference on Mobile Computing and Networking (MOBICOM'00)*, pages 255–265, August 2000.

[135] J.P. Martin-Flatin. *Web-Based Management of IP Networks and Systems*. John Wiley and Sons, 2002.

[136] S. McCanne and V. Jacobson. The BSD Packet Filter : A New Architecture for Level-User Packet Capture. In *Proc. of the Winter 1993 USENIX conference (USENIX'93)*, San Diego, CA, USA, January 1993.

[137] K. McCloghrie, D. Perkins, J. Schoenwaelder, J. Case, M. Rose, and S. Waldbusser. Structure of Management Information Version 2 (SMIv2). http ://www.ietf.org/rfc/rfc2578.txt, January 2000. IETF Request for Comments 2578 (Proposed Standard).

[138] K. McCloghrie and M. Rose. Management Information Base for Network Management of TCP/IP-based internets : MIB-II. http ://www.ietf.org/rfc/rfc1213.txt, March 1991. IETF Request for Comments 1213.

[139] P. Michiardi and R. Molva. Analysis of Coalition Formation and Cooperation Strategies in Mobile Ad-Hoc Networks. *Ad Hoc Networks Journal*, 3(2), March 2005.

[140] D. Middleton. *An Introduction to Statistical Communication Theory*. Wiley-IEEE Press, April 1996.

[141] B. A. Miller and C. Bisdikian. *Bluetooth Revealed*. Prentice Hall Professional, Indianapolis, IN, USA, 2000.

[142] A. Misra, S. Das, A. McAuley, and S. K. Das. Autoconfiguration, Registration, and Mobility Management for Pervasive Computing. *IEEE Personal Communications, Special Issue on Pervasive Computing*, 8(4), August 2001.

[143] M. Mohsin and R. Prakash. IP Address Assignment in a Mobile Ad Hoc Network. In *Proc. of IEEE Military Communications Conference (MILCOM'02)*, volume 2, pages 856–861, October 2002.

[144] J. T. Moy. *OSPF : Anatomy of An Internet Routing Protocol*. Addison-Wesley Professional, January 1998.

[145] A. Munaretto, S. Mauro, P. Fonseca, and N. Agoulmine. Policy-based management of ad-hoc enterprise networks. In *HP Openview University Association 9th Annual Workshop*, June 2002.

[146] R. Murch. *Autonomic Computing*. IBM Press On Demand Series, March 2004.

[147] C. Siva Ram Murthy and B.S. Manoj. *Ad-hoc Wireless Networks : Architectures and Protocols*. Number ISBN 0-13-147023-X. Prentice Hall (Eds.), New Jersey, USA, 2004.

[148] S. Nesargi and R. Prakash. MANETconf : Configuration of Hosts in a Mobile Ad-hoc Network. In *Proc. of the 21st IEEE International Conference on Computer Communications (INFOCOM'02)*, New York, NY, USA, June 2002.

[149] D. Ngo and J. Wu. WANMON : a Resource Usage Monitoring Tool for Ad-hoc Wireless Networks. In *Proc. of the 28th Annual IEEE Conference on Local Computer Networks (LCN'03)*, pages 738–745, Bonn, Germany, October 2003. IEEE Computer Society.

[150] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS) using AOA. In *Proc. of the 22nd IEEE International Conference on Computer Communications (INFOCOM'03)*, San Francisco, CA, USA, April 2003.

[151] L. Novak and A. Gibbons. *Hybrid Graph Theory and Network Analysis*. Cambridge University Press, September 1999.

[152] F. Ohrtman. *WiMAX Handbook - Building 802.16 Wireless Networks*. Mc Graw Hill Professional, Columbus, OH, USA, 2005.

[153] N. H. Vaidya P. Kyasanur. Selfish MAC Layer Misbehavior in Wireless Networks. *IEEE Transactions on Mobile Computing*, 4(5), September 2005.

[154] G. Pavlou, P. Flegkas, S. Gouveris, and A. Liotta. On Management Technologies and the Potential of Web Services. *IEEE Communications, Special Issue on XML-based Management of Networks and Services*, 42(7), July 2004.

[155] G. Pei, M. Gerla, and T. W. Chen. Fisheye State Routing : a Routing Scheme for Ad-Hoc Wireless Networks. In *Proc. of the IEEE International Conference on Communications (ICC'00)*, pages 70–74, New Orleans, LA, USA, June 2000.

[156] C. Perkins, E. Belding-Royer, and S. Das. Ad-hoc On-Demand Distance Vector (AODV) Routing. http ://www.ietf.org/rfc/rfc3561.txt, July 2003. IETF Request for Comments 3561.

[157] C. E. Perkins. *Ad-hoc Networking*. Number ISBN 0-201-30976-9. Pearson Education, Addison-Wesley (Eds.), New Jersey, USA, 2000.

[158] C. E. Perkins, J. T. Malinen, R. Wakikawa, A. Nilsson, and A. J. Tuominen. Internet Connectivity for Mobile Ad-Hoc Networks. *Wireless Communications and Mobile Computing, Special Issue on Mobile Ad Hoc Networking : Research, Trends and Applications*, 2(5), 2002.

[159] D. Perkins and E. McGinnis. *Understanding SNMP MIBs*. Prentice Hall Professional, Indianapolis, IN, USA, 1997.

[160] K. Phanse. *Policy-Based Quality of Service Management in Wireless Ad-hoc Networks*. PhD thesis, Faculty of the Virginia Polytechnic Institute and State University, August 2003.

[161] K. Phanse and L. DaSilva. Addressing the Requirements of QoS Management for Wireless Ad-hoc Networks. *International Journal on Computer Communications*, 26(12) :1263–1273, July 2003.

[162] C. Prehofer and C. Bettstetter. Self-organization in Communication Networks : Principles and Design Paradigms. *IEEE Communications Magazine*, 43 :78–85, July 2005.

[163] A. Rajeswaran and R. Negi. Capacity of Power Constrained Ad-Hoc Networks. In *Proc. of the 23rd IEEE International Conference on Computer Communications (INFOCOM'04)*, Hong Kong, March 2004.

[164] K. Ramachandran, E. Belding-Royer, and K. Almeroth. DAMON : A Distributed Architecture for Monitoring Multi-hop Mobile Networks. In *Proc. of IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, Santa Clara, CA, USA, October 2004.

[165] S. Rangarajan, S. Setia, and S. K. Tripathi. A Fault-Tolerant Algorithm for Replicated Data Management. *IEEE Transactions on Parallel and Distributed Systems*, 6(12), December 1995.

[166] R.Badonnel, L. Andrey, and O. Festor. Composition de Services et Supervision. In *Proc. of Les Nouvelles Technologies de la Répartition (NOTERE'2004)*, Saidia, Maroc, July 2004. Lecture Notes in Computer Science 3278, taux d'acceptation : 22.2%.

[167] G. Ricart and A. K. Agrawala. An Optimal Algorithm for Mutual Exclusion in Computer Networks. *Communications of the ACM*, 24(1), January 1998.

[168] M. Roesch. Snort : Lightweight Intrusion Detection for Networks. http ://www.snort.org.

[169] K. Romer. Time Synchronization in Ad-Hoc Networks. In *Proc. of ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MOBIHOC'00)*, Boston, MA, USA, August 2000.

[170] C. E. Perkins S. Singh and T. Clausen. Ad-Hoc Network Autoconfiguration : Definition and Problem Statement. IETF draft, http ://www.ietf.org/internet-drafts/draft-singh-autoconf-adp-00.txt, February 2004.

[171] S. Schütz, L. Eggert, S. Schmid, and M. Brunner. MANET Simulation Studies : The Incredibles. *Mobile Computing and Communications Review*, 9(4), October 2005.

[172] S. Schütz, L. Eggert, S. Schmid, and M. Brunner. Protocol Enhancements for Intermittently Connected Hosts. *ACM SIGCOMM Computer Communication Review*, 35(3), July 2005.

[173] B. Schwartz, A. W. Jackson, W. T. Strayer, W. Zhou, R. D. Rockwell, and C. Partridge. Smart Packets : Applying Active Networks to Network Management. *ACM Transactions on Computer Systems*, 18, February 2000.

[174] S.Z. Selim and M.A. Ismail. K-Means-Type Algorithms : a Generalized Convergence Theorem and Characterisation of Local Optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1) :81–87, 1984.

[175] C. E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27 :379–423, 1948.

[176] T. Sheldon. *McGraw-Hill's Encyclopedia of Networking and Telecommunications*. McGraw-Hill Companies, May 2001.

[177] C.-C. Shen, C. Jaikaeo, C. Srisathapornphat, and Z. Huang. The GUERRILLA Management Architecture for Ad-hoc Networks. In *Proc. of IEEE Military Communications Conference (MILCOM'02)*, Anaheim, CA, USA, October 2002.

[178] N. Simoni, S. Znaty, and N. Perdigues. *Gestion de Réseau et de Service : Similitude des Concepts, Spécificité des Solutions*. InterEditions, 1997.

[179] D. Simplot-Ryl and I. Stojmenovic, editors. *Special issue on Ad Hoc Networking : Data Communications and Topology Control*, volume 18, July 2004.

[180] F. Sivrikaya and B. Yener. Time Synchronization in Sensor Networks : a Survey. *IEEE Network*, 18(4), July 2004.

[181] C. Smith. *3G Wireless Networks*. Mc Graw Hill Professional, Columbus, OH, USA, 2002.

[182] A. Snow and S. Agarwal. Towards an Optimal Network Survivability Threshold. In *Proc. of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM'05)*, pages 761 – 774, Nice, France, April 2005. IEEE Communications Society.

[183] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. Rao. Cooperation in Wireless Ad-Hoc Networks. In *Proc. of the 22nd IEEE International Conference on Computer Communications (INFOCOM'03)*, San Francisco, CA, USA, April 2003.

[184] D. Subhadrabandhu, S. Devavrat Shah, D. Sarkar, and F. Anjum. A Statistical Framework for Intrusion Detection in Ad Hoc Networks. In *Proc. of the 25th IEEE International Conference on Computer Communications (INFOCOM'06)*, Barcelona Spain, April 2006.

[185] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. http ://www.ietf.org/rfc/rfc2462.txt, December 1998. IETF Request for Comments 2462.

[186] C.K. Toh. Maximum Battery Lifetime Routing to Support Ubiquitous Mobile Computing in Wireless Ad-Hoc Networks. *IEEE Communications Magazine*, 39(6), June 2001.

[187] O. Tomarchio, M. Bisignano, A. Calvagna, and G. Di Modica. ExPeerience : A JXTA Middleware for Mobile Ad-Hoc Networks. In *Proc. of the 3rd International Conference on Peer-to-Peer Computing (P2P'03)*, Linkoping, Sweden, September 2003.

[188] E. van der Vlist. *XML Schema : the W3C Object-Oriented Descriptions for XML*. O'Reilly, June 2002.

[189] W. Vanbenepe and H. Kreger. WSDM 1.0 Standard Specifications. http ://docs.oasis-open.org/wsdm/2004/12/, March 2005. OASIS Web Services Distributed Management TC.

[190] D.C. Verma and D. Verma. *Policy-Based Networking : Architecture and Algorithms*. New Riders Publishers, November 2000.

[191] A. Carneiro Viana, M. Dias de Amorim, S. Fdida, Y. Viniotis, and J. Ferreira de Rezende. Easily-Managed and Topology-Independent Location Service for Self-Organizing Networks. In *Proc. of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'05)*, Urbana-Champaign, IL, USA, May 2005.

[192] B. Wang, S. Sohraab, J. K. Shapiro, and P. N. Tan. Local Detection of Selfish Routing Behavior in Ad Hoc Networks. In *Proc. of the 8th International Symposium on Parallel Architectures,Algorithms and Networks (ISPAN'05)*, Las Vegas, NV, USA, December 2005.

[193] G. Weikum and G. Vossen. *Transactional Information Systems : Theory, Algorithms, and the Practice of Concurrency Control*. Morgan Kaufmann, Series in Data Management Systems, May 2001.

[194] K. Weniger and M. Zitterbart. IPv6 Autoconfiguration in Large Scale Mobile Ad-Hoc Networks. In *Proc. of the European Wireless Conference (European Wireless'02)*, Florence, Italy, February 2002.

[195] C. Westphal. On Maximizing the Lifetime of Distributed Information in Ad-Hoc Networks with Individual Constraints. In *Proc. of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'05)*, Urbana-Champaign, IL, USA, May 2005.

[196] J. Wu and F. Dai. A Distributed Formation of a Virtual Backbone in MANETs Using Adjustable Transmission Ranges. In *Proc. of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS'04)*, Tokyo, Japan, March 2004.

[197] Y. Yemini, G. Goldszmidt, and S. Yemini. Network Management by Delegation. In *Proc. of the 2nd IFIP/IEEE International Symposium on Integrated Network Management (IM'91)*, Washington, DC, USA, 1991. Elsevier North-Holland.

[198] F. Yergeau, T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler. Extensible Markup Language (XML) 1.0 (Third Edition). http ://www.w3.org/TR/2004/REC-xml-20040204/, February 2004. W3C Recommendation.

[199] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In *Proc. of 22nd IEEE International Conference on Computer Communications (INFOCOM'03)*, pages 1312–1321, San Francisco, CA, USA, April 2003.

[200] R. Zhang and M. Bartell. *BGP Design and Implementation*. Cisco Press, December 2003.

[201] Z. Zhang, B. Tian Dai, and A. K.H. Tung. On the Lower Bound of Local Optimums in K-Means Algorithm. In *Proc. of the IEEE International Conference on Datamining (ICDM'06)*, Hong Kong, December 2006.

[202] H. Zhou, L. M. Ni, and M. W. Mutka. Prophet Address Allocation for Large Scale MANETs. *Ad Hoc Networks Journal*, 1(4), November 2003.

[203] H. Zhou, L. NiMatt, and W. Mutka. Prophet Address Allocation for Large Scale MANETs. In *Proc. of the 22nd IEEE International Conference on Computer Communications (INFOCOM'03)*, San Francisco, CA, USA, April 2003.

[204] M.H. Zweig and G. Campbell. Receiver-Operating Characteristic (ROC) Plots : a Fundamental Evaluation Tool. *Clinical Chemistry*, 29(4) :561–577, 1993.

Glossaire

ACPI	Advanced Configuration and Power Interface	40
ANMP	Ad-Hoc Network Management Protocol	27
AODV	Ad-hoc On-Demand Distance Vector	25
ASN	Abstract Syntax Notation	177
BGP	Border Gateway Protocol	64
BPF	Berkeley Packet Filtering	146
CIM	Common Information Model	64
CLTC	Cluster-based Topology Control	33
COPS	Common Open Policy Service	29
COPS-PR	Common Open Policy Service for PRovisioning	29
CPU	Central Processing Unit	29
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance	48
DA	Degré d'Atteignabilité	103
DAMON	Distributed Architecture for Monitoring Mobile Networks	40
DARPA	Defense Advanced Research Projects Agency	9
DCDP	Dynamic Configuration and Distribution Protocol	44
DCF	Distributed Coordination Function	48
DHCP	Dynamic Host Configuration Protocol	43
DMTF	Distributed Management Task Force	64
DPA	Domain Policy Agent	29
FCAPS	Fault Configuration Accounting Performance Security	22
FN	False Negative	133
FP	True Positive	133
GPA	Global Policy Agent	29
HNA	Host and Network Association	77
IBM	International Business Machines	10
ID	Identifiant	68
IEEE	Institute of Electrical and Electronics Engineers	92
IETF	Internet Engineering Task Force	10

INRIA	Institut National de Recherche en Informatique et Automatique	11
IP	Internet Protocol	10
IPv6	Internet Protocol version 6	43
ISO	International Standards Organization	22
JAMA	Java Matrix Package	150
LACM	Level-Access Control Model	28
LORIA	Laboratoire Lorrain de Recherche en Informatique et ses Applications	11
LPA	Local Policy Agent	29
MAC	Medium Access Control	72
MANET	Mobile Ad-Hoc Networks	23
Mbps	Megabits par seconde	112
MID	Multiple Interface Declaration	76
MOF	Managed Object Format	19
MPR	Relais MultiPoints	74
NetConf	Network Configuration Protocol	20
NRL	Navy Research Laboratory	92
ns-2	Network Simulator 2	92
OLSR	Optimized Link State Routing Protocol	74
OSPF	Open Shortest Path First	64
PBM	Policy Based Management	29
PCF	Point Coordination Function	48
PDP	Policy Decision Point	29
PDU	Packet Data Unit	70
PEP	Policy Enforcement Point	29
PR	Participation au Routage	104
PRNET	Packet Radio Network	9
RFC	Request For Comments	25
ROC	Receiver Operating Characteristic	134
RPC	Remote Procedure Call	22
RPGM	Reference Point Group Mobility Model	96
RWP	Random Way Point Mobility Model	92
SMI	Structure of Management Information	19
Sn	Sensibilité	133
SNMP	Simple Network Management Protocol	19
Sp	Spécificité	133
SURAN	Survival Radio Network	10

TC	Topology Control	74
TN	True Negative	133
TP	True Positive	133
UML	Unified Modeling Language	65
WANMON	Wireless Ad-Hoc Network Monitoring Tool	40
WBEM	Web-Based Enterprise Management	64
WiMAX	Worldwide Interoperability for Microwave Access	9
XML	eXtensible Markup Language	20
ZRP	Zone Routing Protocol	25

Annexe A

Spécification de la base d'informations de gestion pour le protocole de routage OLSR

Ce document présente la base d'informations de gestion OLSR-MIB que nous avons proposée pour le protocole de routage OLSR (RFC 3626) [57] et avons spécifiée à l'aide de la notation standard ASN.1[60].

```
-- File Name : OLSR-MIB v1.1
-- Date      : Thu Jan 27 09:24:05 CET 2006
-- Author    : Remi Badonnel, Radu State, Olivier Festor
OLSR-MIB        DEFINITIONS ::= BEGIN
        IMPORTS
                RowStatus, TimeInterval
                        FROM SNMPv2-TC
                OBJECT-GROUP
                        FROM SNMPv2-CONF
                experimental, MODULE-IDENTITY, enterprises, OBJECT-TYPE,
                    Integer32, IpAddress, Unsigned32
                        FROM SNMPv2-SMI;

        olsr    MODULE-IDENTITY
                ORGANIZATION    "LORIA - INRIA Lorraine"
                CONTACT-INFO    "RB RS OF"
                DESCRIPTION             "The MIB module to describe the
                   OLSR protocol (RFC 3626)"
                ::=  {  experimental  1  }

        org     OBJECT IDENTIFIER
                ::=  {  iso  3  }

        dod     OBJECT IDENTIFIER
                ::=  {  org  6  }

        internet        OBJECT IDENTIFIER
                ::=  {  dod  1  }
```

```
experimental    OBJECT IDENTIFIER
        ::=  {  internet  3  }

private OBJECT IDENTIFIER
        ::=  {  internet  4  }

enterprises     OBJECT IDENTIFIER
        ::=  {  private  1  }

olsrGeneralGroup        OBJECT-GROUP
        OBJECTS                 { olsrMainAddr,
           olsrVersionNumber, olsrWillingness, olsrIfTable }
        STATUS                  current
        DESCRIPTION             "The general group."
        ::=  {  olsr  1  }

olsrMainAddr    OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "olsrMainAddr is equivalent of
           router IP or node identifier. A multiple OLSR
           interface node must chosen one of its OLSR interface
           addresses as its main address."
        REFERENCE               "RFC 3626 section 1.1"
        ::=  {  olsr  2  }

olsrVersionNumber       OBJECT-TYPE
        SYNTAX                  Unsigned32
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The OLSR version number."
        ::=  {  olsr  3  }

olsrWillingness OBJECT-TYPE
        SYNTAX                  INTEGER  { willnever ( 0 ) ,
           willlow ( 1 ) , willdefault ( 3 ) , willhigh ( 6 ) ,
           willalways ( 7 ) }
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The willingness of a node
           specifies how willing a node is to be forwarding
           traffic on behalf of other nodes."
        REFERENCE               "RFC 3626 section 18.8"
        DEFVAL                  { willdefault }
        ::=  {  olsr  4  }

olsrMPRCoverage OBJECT-TYPE
        SYNTAX          Unsigned32
        MAX-ACCESS              read-only
        STATUS          current
        DESCRIPTION             "The MPR coverage specifies by
           how many MPR nodes any strict 2-hop node should be
           covered."
```

```
        REFERENCE               "RFC 326 section 16.1"
        DEFVAL                  { 1 }
        ::=  {  olsr  5  }

olsrIfTable     OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrIfEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The list of OLSR interfaces."
        REFERENCE
        "RFC 3626 section 1.1"
        ::=  { olsr  6 }

olsrIfEntry     OBJECT-TYPE
        SYNTAX          OlsrIfEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrIfAddr  }
        ::=  { olsrIfTable 1 }

OlsrIfEntry  ::=  SEQUENCE {
        olsrIfAddr  IpAddress,
        olsrIfRowStatus  RowStatus
        }


olsrIfAddr      OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The OLSR interface's address at
            the IP layer."
        ::=  {  olsrIfEntry  1  }


olsrIfRowStatus OBJECT-TYPE
        SYNTAX                  RowStatus  { active ( 1 ) ,
           notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4
            ) , createAndWait ( 5 ) , destroy ( 6 ) }
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "This object is used to create
           and delete rows in the olsrIfTable."
        ::=  {  olsrIfEntry  3  }

olsrLocalLinkGroup      OBJECT-GROUP
        OBJECTS                 { olsrLinkTable }
        STATUS                  current
        DESCRIPTION             "The local link group stores
           information about links to neighbors."
        REFERENCE               "RFC 3626 section 4.2"
        ::=  {  olsr  7  }
```

```
olsrLinkTable   OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrLinkEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The table to describe the link set."
        REFERENCE
        "RFC 3626 section 4.2.1"
        ::=  { olsr  8 }

olsrLinkEntry   OBJECT-TYPE
        SYNTAX          OlsrLinkEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrLinkNeighborIfAddr, olsrIfAddr  }
        ::=  { olsrLinkTable 1 }

OlsrLinkEntry  ::=  SEQUENCE {
        olsrLinkNeighborIfAddr  IpAddress,
        olsrLinkSymTime  TimeInterval,
        olsrLinkAsymTime  TimeInterval,
        olsrLinkExpirationTime  TimeInterval,
        olsrLinkLostTime  TimeInterval,
        olsrLinkPending  INTEGER,
        olsrLinkQuality  Unsigned32,
        olsrLinkRowStatus  RowStatus
        }


olsrLinkNeighborIfAddr  OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The interface address of the
           neighbor node."
        REFERENCE               "L_neighbor_iface_addr"
        ::=  {  olsrLinkEntry  2  }


olsrLinkSymTime OBJECT-TYPE
        SYNTAX                  TimeInterval  ( -2147483648 ..
           2147483647  )
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The time until which the link
           is considered symmetric."
        REFERENCE               "L_SYM_time"
        ::=  {  olsrLinkEntry  3  }


olsrLinkAsymTime        OBJECT-TYPE
        SYNTAX                  TimeInterval  ( -2147483648 ..
           2147483647  )
        MAX-ACCESS              read-only
```

```
        STATUS                  current
        DESCRIPTION             "The time until which the
           neighbor interface is considered heard."
        REFERENCE               "L_ASYM_time"
        ::=  {  olsrLinkEntry  4  }


olsrLinkExpirationTime  OBJECT-TYPE
        SYNTAX                  TimeInterval  ( -2147483648 ..
           2147483647  )
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The time at which this record
           expires and must be removed."
        REFERENCE               "L_time"
        ::=  {  olsrLinkEntry  5  }


olsrLinkLostTime        OBJECT-TYPE
        SYNTAX                  TimeInterval  ( -2147483648 ..
           2147483647  )
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "A timer for declaring a link as
            lost when an established link becomes pending.
           OPTIONAL"
        REFERENCE               "RFC 3626 section 14.1"
        ::=  {  olsrLinkEntry  6  }


olsrLinkPending OBJECT-TYPE
        SYNTAX                  INTEGER  { false ( 0 ) , true (
           1 ) }
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "A boolean to specify if the
           link is considered pending. OPTIONAL"
        REFERENCE               "RFC 3626 section 14.1"
        ::=  {  olsrLinkEntry  7  }


olsrLinkQuality OBJECT-TYPE
        SYNTAX                  Unsigned32
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The inverse value of a
           dimension less number between 0 and 1 describing the
           quality of the link. OPTIONAL"
        REFERENCE               "RFC 3626 section 14.1"
        ::=  {  olsrLinkEntry  8  }


olsrLinkRowStatus       OBJECT-TYPE
```

```
        SYNTAX                  RowStatus  { active ( 1 ) ,
           notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4
            ) , createAndWait ( 5 ) , destroy ( 6 ) }
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "This object is used to create
           and delete rows in the olsrLinkTable."
        ::=  {  olsrLinkEntry  9  }

olsrNeigborhoodGroup    OBJECT-GROUP
        OBJECTS                 { olsrNbrTable, olsr2hNbrTable,
           olsrMPRTable, olsrMPRSelectorTable }
        STATUS                  current
        DESCRIPTION             "The neighborhood group stores
           information about neighbors, 2-hop neighbors, MPRs
           and MPR selectors."
        REFERENCE               "RFC 3626 section 4.3"
        ::=  {  olsr  9  }

olsrNbrTable    OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrNbrEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The neighbor set."
        REFERENCE
        "RFC 3626 section 4.3.1"
        ::=  { olsr  10 }

olsrNbrEntry    OBJECT-TYPE
        SYNTAX          OlsrNbrEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrNbrMainAddr  }
        ::=  { olsrNbrTable 1 }

OlsrNbrEntry  ::=  SEQUENCE {
        olsrNbrMainAddr  IpAddress,
        olsrNbrStatus  INTEGER,
        olsrNbrWillingness  INTEGER,
        olsrNbrRowStatus  RowStatus
        }


olsrNbrMainAddr OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The main address of a neigbhor
           ."
        REFERENCE               "N_neighbor_main_addr"
        ::=  {  olsrNbrEntry  1  }
```

```
olsrNbrStatus   OBJECT-TYPE
        SYNTAX                  INTEGER  { notsym ( 0 ) }
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "An integer to specify if the
           neighbor is symmetric."
        REFERENCE               "N_Status"
        ::=  {  olsrNbrEntry  2  }


olsrNbrWillingness      OBJECT-TYPE
        SYNTAX                  INTEGER  { willnever ( 0 ) ,
           willlow ( 1 ) , willdefault ( 3 ) , willhigh ( 4 ) ,
           willalways ( 7 ) }
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "An integer between 0 and 7 to
           specify a nodes willingness to carry traffic on
           behalf of other nodes."
        REFERENCE               "N_willingness"
        DEFVAL                  { willdefault }
        ::=  {  olsrNbrEntry  3  }


olsrNbrRowStatus        OBJECT-TYPE
        SYNTAX                  RowStatus  { active ( 1 ) ,
           notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4
            ) , createAndWait ( 5 ) , destroy ( 6 ) }
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "The object is used to create or
            delete rows in table olsrNbrTable."
        ::=  {  olsrNbrEntry  4  }

olsr2hNbrTable  OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  Olsr2hNbrEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The 2-hop neighbor set."
        REFERENCE
        "RFC 3626 section 4.3.2"
        ::=  { olsr  11 }

olsr2hNbrEntry  OBJECT-TYPE
        SYNTAX          Olsr2hNbrEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrNbrMainAddr, olsr2hNbrMainAddr  }
        ::=  { olsr2hNbrTable 1 }

Olsr2hNbrEntry  ::=  SEQUENCE {
        olsr2hNbrMainAddr  IpAddress,
        olsr2hNbrExpirationTime  TimeInterval,
```

```
        olsr2hNbrRowStatus  RowStatus
        }


olsr2hNbrMainAddr       OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The main address of a 2-hop
           neighbor node."
        REFERENCE               "N_neighbor_main_addr"
        ::=  {  olsr2hNbrEntry  1  }


olsr2hNbrExpirationTime OBJECT-TYPE
        SYNTAX                  TimeInterval
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The time at which a record
           expires and must be removed."
        REFERENCE               "N_time"
        ::=  {  olsr2hNbrEntry  2  }


olsr2hNbrRowStatus      OBJECT-TYPE
        SYNTAX                  RowStatus  { active ( 1 ) ,
           notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4
            ) , createAndWait ( 5 ) , destroy ( 6 ) }
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "The object is used to create
           and delete rows in table olsr2hNbrTable."
        ::=  {  olsr2hNbrEntry  3  }

olsrMPRTable    OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrMPREntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The MPR set."
        REFERENCE
        "RFC 3626 section 4.3.3"
        ::=  { olsr  12 }

olsrMPREntry    OBJECT-TYPE
        SYNTAX          OlsrMPREntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrMPRMainAddr  }
        ::=  { olsrMPRTable 1 }

OlsrMPREntry  ::=  SEQUENCE {
        olsrMPRMainAddr  IpAddress,
        olsrMPRRowStatus  RowStatus
```

```
        }


olsrMPRMainAddr OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The main address of a neighbor
           selected as MPR."
        ::=  {  olsrMPREntry  1  }


olsrMPRRowStatus        OBJECT-TYPE
        SYNTAX                  RowStatus  { active ( 1 ) ,
           notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4
            ) , createAndWait ( 5 ) , destroy ( 6 ) }
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "The object is used to create
           and delete rows in the olsrMPRTable."
        ::=  {  olsrMPREntry  2  }

olsrMPRSelectorTable    OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrMPRSelectorEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The MPR selector set."
        REFERENCE
        "RFC 3626 section 4.3.4"
        ::=  { olsr  13 }

olsrMPRSelectorEntry    OBJECT-TYPE
        SYNTAX          OlsrMPRSelectorEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrMPRSelectorMainAddr  }
        ::=  { olsrMPRSelectorTable 1 }

OlsrMPRSelectorEntry  ::=  SEQUENCE {
        olsrMPRSelectorMainAddr  IpAddress,
        olsrMPRSelectorExpirationTime  TimeInterval,
        olsrMPRSelectorRowStatus  RowStatus
        }


olsrMPRSelectorMainAddr OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The main address of a MPR
           selector node."
        REFERENCE               "MS_main_addr"
        ::=  {  olsrMPRSelectorEntry  1  }
```

```
olsrMPRSelectorExpirationTime   OBJECT-TYPE
        SYNTAX                  TimeInterval
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The time at which a record
           expires and must be removed."
        REFERENCE               "MS_time"
        ::=  {  olsrMPRSelectorEntry  2  }


olsrMPRSelectorRowStatus        OBJECT-TYPE
        SYNTAX                  RowStatus
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "The object is used to create
           and delete rows in olsrMRPSelectorTable."
        ::=  {  olsrMPRSelectorEntry  3  }

olsrTopologyGroup       OBJECT-GROUP
        OBJECTS                 { olsrTopTable, olsrMIDTable,
           olsrHNATable }
        STATUS                  current
        DESCRIPTION             "The topology group stores
           information on the network topology and is used for
           routing table calculations."
        REFERENCE               "RFC 3626 section 4.4"
        ::=  {  olsr  14  }

olsrTopTable    OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrTopEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The topology table."
        REFERENCE
        "RFC 3626 section 4.4"
        ::=  { olsr  15 }

olsrTopEntry    OBJECT-TYPE
        SYNTAX          OlsrTopEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrTopDestMainAddr,
           olsrTopLastMainAddr  }
        ::=  { olsrTopTable 1 }

OlsrTopEntry  ::=  SEQUENCE {
        olsrTopDestMainAddr  IpAddress,
        olsrTopLastMainAddr  IpAddress,
        olsrTopSequence  Unsigned32,
        olsrTopExpirationTime  TimeInterval,
        olsrTopRowStatus  RowStatus
```

```
        }


olsrTopDestMainAddr     OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The main address of a node,
           which may be reached in one hop from the node with
           the main address olsrLastMainAddr."
        REFERENCE               "T_dest_addr"
        ::=  {  olsrTopEntry  1  }


olsrTopLastMainAddr     OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "Typically, the main address of
           a MPR of olsrDestMainAddr."
        REFERENCE               "L_last_addr"
        ::=  {  olsrTopEntry  2  }


olsrTopSequence OBJECT-TYPE
        SYNTAX                  Unsigned32
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The sequence number."
        REFERENCE               "T_seq"
        ::=  {  olsrTopEntry  3  }


olsrTopExpirationTime   OBJECT-TYPE
        SYNTAX                  TimeInterval
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The time at which the record
           expires and must be removed."
        REFERENCE               "T_time"
        ::=  {  olsrTopEntry  4  }


olsrTopRowStatus        OBJECT-TYPE
        SYNTAX                  RowStatus
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "The object is used to create
           and delete rows in the olsrTopTable."
        ::=  {  olsrTopEntry  5  }

olsrMIDTable    OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrMIDEntry
        MAX-ACCESS      not-accessible
```

```
        STATUS          current
        DESCRIPTION     "The table describes the multiple
           interface associations."
        REFERENCE
        "RFC 3626 section 5"
        ::=  { olsr  16 }

olsrMIDEntry    OBJECT-TYPE
        SYNTAX          OlsrMIDEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrTopDestMainAddr, olsrMIDMainAddr
            }
        ::=  { olsrMIDTable 1 }

OlsrMIDEntry  ::=  SEQUENCE {
        olsrMIDMainAddr  IpAddress,
        olsrMIDExpirationTime  TimeInterval,
        olsrMIDRowStatus  RowStatus
        }


olsrMIDMainAddr OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The main address associates
           with the interface address."
        REFERENCE               "I_iface_addr"
        ::=  {  olsrMIDEntry  1  }


olsrMIDExpirationTime   OBJECT-TYPE
        SYNTAX                  TimeInterval  ( -2147483648 ..
           2147483647  )
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The time at which the records
           expires and must be removed."
        REFERENCE               "I_time"
        ::=  {  olsrMIDEntry  2  }


olsrMIDRowStatus        OBJECT-TYPE
        SYNTAX                  RowStatus  { active ( 1 ) ,
           notInService ( 2 ) , notReady ( 3 ) , createAndGo ( 4
            ) , createAndWait ( 5 ) , destroy ( 6 ) }
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "The object is used to create
           and delete rows in the olsrMIDTable."
        ::=  {  olsrMIDEntry  3  }
```

```
olsrHNATable    OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrHNAEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The table of host and network
           associations."
        REFERENCE
        "RFC 3626 section 12.2"
        ::=  { olsr  17 }

olsrHNAEntry    OBJECT-TYPE
        SYNTAX          OlsrHNAEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrHNANetAddr, olsrHNAGatewayAddr  }
        ::=  { olsrHNATable 1 }

OlsrHNAEntry  ::=  SEQUENCE {
        olsrHNAGatewayAddr  IpAddress,
        olsrHNANetAddr  IpAddress,
        olsrHNANetmask  IpAddress,
        olsrHNAExpirationTime  TimeInterval,
        olsrHNARowStatus  RowStatus
        }


olsrHNAGatewayAddr      OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The address of a OLSR interface
            of a gateway."
        REFERENCE               "A_gateway_addr"
        ::=  {  olsrHNAEntry  1  }


olsrHNANetAddr  OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              not-accessible
        STATUS                  current
        DESCRIPTION             "The network address of a
           network reachable through this gateway."
        REFERENCE               "A_network_addr"
        ::=  {  olsrHNAEntry  2  }


olsrHNANetmask  OBJECT-TYPE
        SYNTAX                  IpAddress
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The netmask of a network
           reachable through this gateway."
        REFERENCE               "A_netmask"
```

```
        ::=  {  olsrHNAEntry  3  }


olsrHNAExpirationTime   OBJECT-TYPE
        SYNTAX                  TimeInterval
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The time at which the record
           expires and must be removed."
        REFERENCE               "A_time"
        ::=  {  olsrHNAEntry  4  }


olsrHNARowStatus        OBJECT-TYPE
        SYNTAX                  RowStatus
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "The object is used to create
           and delete rows in the olsrHNATable."
        ::=  {  olsrHNAEntry  5  }

olsrRoutingGroup        OBJECT-GROUP
        OBJECTS                 { olsrRoutingTable }
        STATUS                  current
        DESCRIPTION             "The routing group stores
           information about the routing table."
        REFERENCE               "RFC 3626 section 10"
        ::=  {  olsr  18  }

olsrRoutingTable        OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrRoutingEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The OLSR routing table."
        REFERENCE
        "RFC 3626 section 10"
        ::=  { olsr  19 }

olsrRoutingEntry        OBJECT-TYPE
        SYNTAX          OlsrRoutingEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrTopDestMainAddr, olsrNbrMainAddr,
            olsrIfAddr  }
        ::=  { olsrRoutingTable 1 }

OlsrRoutingEntry  ::=  SEQUENCE {
        olsrRoutingDist  Unsigned32,
        olsrRoutingRowStatus  RowStatus
        }


olsrRoutingDist OBJECT-TYPE
```

```
        SYNTAX                  Unsigned32
        MAX-ACCESS              read-only
        STATUS                  current
        DESCRIPTION             "The node identified by
           olsrTopDestMainAddr is estimated to be
           olsrRoutingTableDist away from the local node."
        REFERENCE               "R_dist"
        ::=  {  olsrRoutingEntry  1  }


olsrRoutingRowStatus    OBJECT-TYPE
        SYNTAX                  RowStatus
        MAX-ACCESS              read-create
        STATUS                  current
        DESCRIPTION             "The object is used to create
           and delete rows in the olsrRoutingTable."
        ::=  {  olsrRoutingEntry  2  }

olsrMsgGroup    OBJECT-GROUP
        OBJECTS                 { olsrDuplicateTable }
        STATUS                  current
        DESCRIPTION             "The message group stores
           information on the routing messages (HELLO, HELLO,
           HNA, MID)."
        ::=  {  olsr  20  }

olsrDuplicateTable      OBJECT-TYPE
        SYNTAX          SEQUENCE  OF  OlsrDuplicateEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "The duplicate set."
        REFERENCE
        "RFC 3626 section 3.4"
        ::=  { olsr  21 }

olsrDuplicateEntry      OBJECT-TYPE
        SYNTAX          OlsrDuplicateEntry
        MAX-ACCESS      not-accessible
        STATUS          current
        DESCRIPTION     "Row Description"
        INDEX           {  olsrIfAddr, olsrDuplicateOrigMainAddr
           , olsrDuplicateSequence  }
        ::=  { olsrDuplicateTable 1 }

OlsrDuplicateEntry  ::=  SEQUENCE {
        olsrDuplicateOrigMainAddr  IpAddress,
        olsrDuplicateSequence  Unsigned32,
        olsrDuplicateRetransmitted  INTEGER,
        olsrDuplicateExpirationTime  TimeInterval,
        olsrDuplicateRowStatus  RowStatus
        }


olsrDuplicateOrigMainAddr       OBJECT-TYPE
```

```
                SYNTAX                  IpAddress
                MAX-ACCESS              not-accessible
                STATUS                  current
                DESCRIPTION             "The originator main address of
                   the message."
                REFERENCE               "D_addr"
                ::=  {  olsrDuplicateEntry  1  }


        olsrDuplicateSequence   OBJECT-TYPE
                SYNTAX                  Unsigned32
                MAX-ACCESS              not-accessible
                STATUS                  current
                DESCRIPTION             "The sequence number of the
                   message."
                ::=  {  olsrDuplicateEntry  2  }


        olsrDuplicateRetransmitted      OBJECT-TYPE
                SYNTAX                  INTEGER  { false ( 0 ) , true (
                   1 ) }
                MAX-ACCESS              read-only
                STATUS                  current
                DESCRIPTION             "A boolean indicating wether the
                    message has been already retransmitted."
                REFERENCE               "D_retransmitted"
                ::=  {  olsrDuplicateEntry  3  }


        olsrDuplicateExpirationTime     OBJECT-TYPE
                SYNTAX                  TimeInterval
                MAX-ACCESS              read-only
                STATUS                  current
                DESCRIPTION             "The time at which the record
                   expires and must be removed."
                ::=  {  olsrDuplicateEntry  4  }


        olsrDuplicateRowStatus  OBJECT-TYPE
                SYNTAX                  RowStatus
                MAX-ACCESS              read-create
                STATUS                  current
                DESCRIPTION             "The object is used to create
                   and delete rows in the olsrDuplicateTable."
                ::=  {  olsrDuplicateEntry  5  }

END
```