

RÉSEAUX DE CAPTEURS SANS FIL À MULTIPLES PILES PROTOCOLAIRES

HABILITATION À DIRIGER DES RECHERCHES DE L'UNIVERSITÉ BLAISE PASCAL

Spécialité Informatique

Présentée par
Alexandre GUITTON

Soutenue le 26 mars 2014

Azzedine BOUKERCHE, Professeur, Université d'Ottawa	<i>Rapporteur</i>
Andrzej DUDA, Professeur, Université de Grenoble	<i>Rapporteur</i>
Pascale MINET, CR HDR, Inria Rocquencourt	<i>Examinatrice</i>
Michel MISSON, Professeur, Université d'Auvergne	<i>Tuteur</i>
Guy PUJOLLE, Professeur, Université Pierre et Marie Curie	<i>Rapporteur</i>
Alain QUILLIOT, Professeur, Université Blaise Pascal	<i>Examineur</i>
Thierry VAL, Professeur, Université de Toulouse	<i>Examineur</i>

Remerciements

Cette habilitation à diriger des recherches en informatique est très importante pour moi, et marque une étape importante pour moi. J'aimerais remercier tous ceux qui m'ont aidé (de manière directe ou indirecte) dans la réalisation des travaux dont parle ce manuscrit. J'essaierais d'être bref pour ne pas alourdir davantage ce manuscrit, et pour ne pas fatiguer le lecteur curieux qui commencerait la lecture du manuscrit par ces quelques lignes.

Je tiens tout d'abord à remercier les membres de mon jury d'avoir accepté d'examiner mes travaux. Plus particulièrement, je remercie Azzedine BOUKERCHE, Andrzej DUDA et Guy PUJOLLE d'avoir accepté d'être rapporteurs, d'avoir relu mes travaux et de m'avoir fait de nombreux commentaires intéressants. Je remercie Pascale MINET d'avoir accepté d'être la présidente du jury. Je remercie Thierry VAL de m'avoir conseillé suite à la lecture d'une version préliminaire du manuscrit. Je remercie Alain QUILLIOT de m'avoir encouragé à soutenir mon habilitation. Finalement, je remercie Michel MISSON, le responsable de l'équipe de recherche dans laquelle je travaille, d'avoir accepté d'être mon tuteur.

Je tiens aussi à remercier toutes les personnes qui ont travaillé avec moi. Pour la recherche, cela inclut évidemment mes co-auteurs, mes futurs co-auteurs et les étudiants que j'ai encadré. Pour l'enseignement, cela inclut mes collègues (notamment du département maths-info de l'UFR Sciences et Technologies, des autres départements de l'UFR ST, et de l'ISIMA) et les étudiants que j'ai eus.

Je tiens finalement remercier tous ceux qui m'entourent dans mon quotidien, mes amis (qui se reconnaîtront tous individuellement, dont beaucoup font partie de Sambagogo ou de N'tra Na Roda), et bien sûr ma famille.

Merci à tous !

Abstract

Wireless sensor networks (WSNs) can address the needs of many monitoring applications, due to their low cost, wireless capabilities and ease of deployment. However, applications that are deployed on WSNs often have various quality of service requirements, and it is difficult for a single network stack to fulfill them. Network architectures where several network stacks coexist are starting to emerge.

In this document, we focus on the interactions between the stacks of a WSN by taking two projects as examples. In the first project (on industrial applications), the optimisation of the protocols lead us to allow interactions between stacks. We generalize this approach by introducing local multi-stack architectures, where the choice of the stack to use is made locally by each node. In the second project (on environmental applications), it is the characteristics of the monitored phenomenon that raises the need of interactions between stacks. We introduce global multi-stack architectures, where the choice of the stack is imposed to nodes. In both projects, we propose solutions to avoid routing loops.

The research works described in this document have been made in collaboration with three PhD students, two of them having already received their PhD. This document also gives several perspectives.

Keywords: wireless sensor networks, routing protocols, MAC protocols, multi-stack architectures.

Résumé

Les réseaux de capteurs sans fil permettent de répondre aux besoins de nombreuses applications de surveillance, en raison de leur faible coût, de leur capacité à communiquer sans fil et de leur facilité de déploiement. Toutefois, les applications déployées sur les réseaux de capteurs sans fil ont souvent des besoins variés de qualités de service, et il est difficile pour une seule pile protocolaire de répondre à ces besoins. Des architectures réseaux dans lesquelles plusieurs piles protocolaires cohabitent commencent à émerger.

Dans ce document, nous nous intéressons aux interactions entre les piles protocolaires d'un réseau de capteurs sans fil, en prenant deux projets pour exemples. Dans le premier projet, qui concerne le domaine industriel, l'optimisation des protocoles proposés nous a amené à autoriser des interactions entre piles protocolaires. Nous généralisons cette approche en introduisant le concept d'architecture multi-piles locale, où le choix de la pile à utiliser est décidé localement par chaque nœud. Dans le second projet, qui concerne le domaine environnemental, ce sont les caractéristiques des phénomènes observés qui déclenchent les interactions entre piles protocolaires. Nous introduisons alors le concept d'architecture multi-piles globale, où le choix d'utiliser une pile ou une autre est imposé aux nœuds. Dans ces deux projets, nous proposons des solutions pour éviter les boucles de routage.

Les travaux décrits dans ce document ont été menés en collaboration avec trois étudiants en thèse, deux ayant déjà soutenu. Ce document décrit aussi les perspectives de ces travaux.

Mots-clés : réseaux de capteurs sans fil, protocoles de routage, protocoles MAC, architectures multi-piles.

TABLE DES MATIÈRES

I Synthèse scientifique	11
1 Introduction et contexte des travaux	13
1.1 Contexte des travaux	14
1.2 Plan	15
2 État de l'art	17
2.1 Architectures mono-piles et protocoles associés	18
2.1.1 Protocoles de la sous-couche MAC	19
2.1.2 Protocoles de la couche réseau	21
2.1.3 Résumé des architectures mono-piles	23
2.2 Architectures multi-piles et protocoles associés	23
2.2.1 Normes IEEE 802.15.4 et ZigBee	24
2.2.2 Autres architectures multi-piles	27
2.2.3 Limites des architectures multi-piles	32
2.2.4 Résumé des architectures multi-piles	33
2.3 Bilan	33
3 Décision locale de changement de piles	35
3.1 Étude d'un cas concret : le projet Ocari	36
3.1.1 Objectifs	36
3.1.2 Sous-couche MAC d'Ocari	37
3.1.3 Couche réseau d'Ocari	59
3.1.4 Synthèse des protocoles réseau et MAC dans Ocari	61
3.2 Architectures multi-piles simples et locales	62
3.2.1 Description des architectures simples	62
3.2.2 Description des architectures locales	65

3.3	Évitement des boucles entre protocoles de routage	69
3.3.1	Protocoles compatibles	70
3.3.2	Protocoles retardables	77
3.4	Conclusions	81
4	Décision globale de changement de piles	83
4.1	Pré-étude d'un cas concret : le projet Clervolc	84
4.1.1	Objectifs de la surveillance par réseau de capteurs sans fil . . .	84
4.1.2	Couche physique pour le projet Clervolc	89
4.1.3	Sous-couche MAC pour le projet Clervolc	91
4.1.4	Couche réseau pour le projet Clervolc	98
4.1.5	Bilan concernant le projet Clervolc	99
4.2	Architectures multi-piles globales	100
4.2.1	États d'un nœud	101
4.2.2	États du réseau	101
4.2.3	Protocoles qui supportent le changement d'état	103
4.2.4	Bilan des architectures globales	107
4.3	Étude de l'évolution du réseau	108
4.3.1	Étude du graphe d'évolution	108
4.3.2	Algorithmes concernant l'évolution	117
4.3.3	Bilan de l'étude sur l'évolution du réseau	125
4.4	Conclusions	126
5	Résumé et perspectives	129
5.1	Résumé des contributions présentées	129
5.2	Résumé des contributions non présentées	130
5.2.1	Démarche de simulation réaliste	130
5.2.2	Routage tolérant au délai	132
5.2.3	Réduction de la congestion	132
5.3	Bilan des encadrements	133
5.4	Perspectives	133
5.4.1	Surveillance environnementale	133
5.4.2	Architectures multi-piles	136
5.4.3	Étude du graphe d'évolution	137
	Références	139
II	Synthèse de mes activités d'enseignant-chercheur	151
6	Parcours	153
7	Production scientifique	155
7.1	Tableau récapitulatif	155
7.2	Indicateurs bibliométriques	156
7.3	Liste complète	157

8	Activités d'encadrement	163
8.1	Co-encadrement de doctorants	163
8.1.1	Co-encadrement de Nassima HADID	164
8.1.2	Co-encadrement de Nancy EL RACHKIDY	164
8.1.3	Co-encadrement de Affoua Thérèse ABY	165
8.2	Encadrement de stages de master recherche	165
8.2.1	Encadrement de Hamid ATTALAH	166
8.2.2	Encadrement de Hani MAHMOUDI	166
8.2.3	Encadrement de Michael ABDALLAH	166
8.2.4	Encadrement de Rezzag ABDERRAHMEN	167
8.3	Réflexions sur ma méthode d'encadrement	167
9	Gestion de la recherche	169
9.1	Participation à des projets	169
9.1.1	Surveillance de routes (2005-2007)	169
9.1.2	Surveillance de sites industriels (2007-2011)	171
9.1.3	Surveillance de sites environnementaux (2012-*)	172
9.2	Communautés scientifiques	173
9.2.1	Collaborations	173
9.2.2	Organisation de manifestations scientifiques	174
9.2.3	Comités scientifiques de conférences (hors organisation)	174
9.2.4	Relectures (hors comités scientifiques)	175
9.2.5	Jurys de thèse (hors co-direction)	176
9.3	Autres activités ayant un lien avec la recherche	176
10	Activités pédagogiques	177
10.1	Liste des enseignements	177
10.1.1	Cours de réseau et de système	177
10.1.2	Cours de programmation	178
10.1.3	Autres cours	178
10.2	Activités administratives liées à l'enseignement	178
10.3	Activités annexes liées à l'enseignement	179
III	Annexes	181
A	Compléments sur le graphe d'évolution	183
A.1	Compléments sur la structure du graphe d'évolution	183
A.2	Propriétés du graphe d'évolution	184
A.3	Compléments sur les sup-irréductibles	188
B	Lexique des abréviations et sigles utilisés	191

Première partie
Synthèse scientifique

CHAPITRE

1

INTRODUCTION ET CONTEXTE DES TRAVAUX

Les applications de surveillance de sites industriels et environnementaux ont pris beaucoup d'ampleur depuis quelques années. La surveillance de tels sites est faite selon plusieurs objectifs. Par exemple, certaines applications visent à améliorer la sécurité des utilisateurs ou du personnel, en surveillant la structure d'édifices (comme la surveillance de la structure d'un pont [HSB10, KPC⁺07]) ou les événements dangereux (comme la propagation d'un feu de forêt [YWM05, HB07]). D'autres applications sont proposées pour améliorer le processus de suivi des données, dans un but économique (comme la surveillance des places libres dans un parking [SSDQ09]) ou écologique (comme la cartographie régulière de la pollution d'une rivière [HUM⁺11, HC12] ou les phénomènes migratoires [DEM⁺09, GSGSL⁺10]). Enfin, des applications visent à améliorer la compréhension de phénomènes naturels (comme par exemple, les glissements de terrain [Sud10]).

Dans les applications de surveillance de sites industriels et environnementaux, on retrouve des problématiques communes. Tout d'abord, les sites à surveiller sont de grande étendue. Il peut s'agir par exemple d'un bâtiment, d'une route, d'un volcan ou d'une forêt. De plus, les sites à surveiller ne sont pas toujours propices à l'installation de moyens d'alimentation électrique (tels que des câbles électriques ou des capteurs solaires par exemple). Pour les sites industriels (comme les centrales nucléaires, les barrages, les routes), l'installation d'une telle infrastructure nécessite des travaux lourds et coûteux, étant donné que le site n'a généralement pas été prévu pour un tel réseau *a priori*. Pour les sites environnementaux, l'installation d'une telle infrastructure n'est généralement pas possible, en raison de l'éloignement entre les sites surveillés et les zones d'habitation, ou en raison des conditions inhospitalières des sites.

Les réseaux de capteurs sans fil représentent une solution adaptée à la surveillance de tels sites. Un réseau de capteurs sans fil est constitué de nœuds (dits nœuds capteurs) de petite taille, à bas prix et visant une autonomie énergétique. Les nœuds capteurs ont des capacités de mesures (par l'intermédiaire des capteurs), de stockage, de traitement des données et de transmissions sans fil. Au sein d'un réseau de capteurs sans fil, les nœuds capteurs effectuent des mesures et collaborent pour les transmettre à des puits, qui sont des nœuds particuliers ayant des capacités importantes de stockage et de ressources énergétiques. Une fois arrivées dans le puits, les données issues des capteurs sont disponibles pour des applications centralisées.

Les avantages des réseaux de capteurs sans fil sont nombreux. Comme les nœuds capteurs visent à être autonomes en énergie, ils peuvent être déployés relativement facilement. Leur faible coût unitaire permet d'envisager un déploiement à grande échelle (c'est-à-dire de quelques centaines de nœuds) afin de surveiller des sites étendus. Leur petite taille leur permet d'être déployés dans un site industriel ou environnemental, sans impacter le processus surveillé de manière significative.

En revanche, les nœuds capteurs souffrent de limitations importantes. La nécessité d'économie d'énergie des nœuds impose que le microprocesseur qu'ils embarquent soit de faible puissance et que leur module radio ne permette des transmissions sans fil qu'à courte portée. La petite taille des nœuds et leur coût réduit imposent que la capacité de stockage des nœuds soit limitée. Toutes ces contraintes complexifient la conception et l'implémentation de protocoles pour les réseaux de capteurs sans fil. Tout au long de ce document, nous tenterons de garder à l'esprit ces contraintes.

1.1 Contexte des travaux

L'utilisation de réseaux de capteurs sans fil pour effectuer de la surveillance pose de nombreux défis. Par exemple, au niveau matériel, il est difficile de concevoir des nœuds capteurs de petite taille et résistant aux intempéries, ou intégrant des sources d'énergie efficaces. Au niveau logiciel, il est difficile de développer des applications ayant une empreinte mémoire limitée et ne consommant que peu de ressources. Au niveau protocolaire, il est difficile de développer des mécanismes de communication fournissant des garanties de qualité de service, ou des mécanismes économes en énergie.

Dans nos travaux de recherche, nous nous concentrons sur les défis protocolaires, en essayant de prendre en compte les contraintes matérielles et logicielles des nœuds capteurs. Plus précisément, nous nous concentrons sur les protocoles de la couche liaison de données (principalement ceux de la sous-couche MAC) et de la couche réseau, ainsi que sur l'architecture générale de la solution. Les protocoles que nous proposons sont souvent étudiés analytiquement, et systématiquement évalués par l'intermédiaire de simulations qui suivent une démarche visant le rapprochement avec la réalité. Nous avons fait le choix de ne pas présenter beaucoup de résultats de simulation dans ce document, afin de ne pas alourdir sa présentation, mais de faire référence à ces résultats dans le texte lorsqu'ils sont publiés¹. Par conséquent,

1. Pour être différenciées des références externes, les références à mes travaux sont suivies d'une

la démarche de simulation que nous avons adoptée n'a pas une place centrale dans ce document, même si elle a une place centrale dans nos travaux de recherche².

Les applications que nous considérons sont de deux types : il s'agit soit de surveillance industrielle, soit de surveillance environnementale. L'originalité des applications que nous considérons (et des solutions que nous proposons) vient principalement des conditions de déploiement particulières pour le réseau de capteurs (dues aux distances de plusieurs dizaines de mètres en moyenne entre les nœuds, à la présence de métal ou de végétation, à l'absence de lignes de vue entre les nœuds, ou encore au déploiement extérieur sujet aux intempéries). Les objectifs que nous nous fixons sont habituels, puisque nous cherchons généralement à réduire le délai et le taux de pertes des paquets, à augmenter le débit du réseau, et ainsi à respecter une certaine qualité de service.

1.2 Plan

Cette synthèse scientifique se concentre sur l'étude d'architectures ayant plusieurs piles protocolaires. L'objectif principal de ces architectures est de diminuer le coût lié au déploiement des réseaux de capteurs sans fil, en faisant fonctionner plusieurs applications au-dessus d'un même déploiement, ce qui revient à mutualiser les nœuds capteurs (ainsi qu'une partie du logiciel des applications), ou dit autrement à partager le réseau de capteurs sans fil. Les multiples applications déployées sur le réseau engendrent des besoins en qualité de service variés, ce qui entraîne des contraintes importantes sur les protocoles de communication. Lorsqu'un seul protocole ne permet pas de fournir les qualités de service attendues [YM10, YIE11], il faut envisager d'intégrer plusieurs protocoles de communication assurant les mêmes fonctionnalités (mais en les remplissant différemment), ce qui conduit aux architectures ayant plusieurs piles protocolaires.

Dans le chapitre 2, nous rappelons les notions de base autour des piles protocolaires. Dans la partie 2.1, nous détaillons quelques protocoles (de la sous-couche MAC et de la couche réseau) utilisés dans les réseaux de capteurs sans fil. Nous montrons qu'il est difficile pour ces protocoles de fournir des qualités de service très variées. Dans la partie 2.2, nous détaillons plusieurs protocoles (toujours de la sous-couche MAC et de la couche réseau) correspondant au concept d'architecture multi-piles. Nous décrivons aussi l'architecture IEEE 802.15.4/ZigBee, qui est à l'heure actuelle une architecture incontournable dans les réseaux de capteurs sans fil, et nous montrons que cette architecture peut être considérée comme multi-piles.

Dans le chapitre 3, nous partons d'un projet de surveillance de site industriel, nommé Ocari, pour introduire la nécessité des architectures multi-piles. Dans la partie 3.1, nous décrivons notre démarche lors de ce projet, qui consiste à surveiller une centrale nucléaire (notamment), ce qui impose des garanties sur la livraison et le délai des transmissions de certaines trames. Nous avons donc développé une architecture pouvant gérer deux types de communications (avec ou sans contrainte de délai), en utilisant alternativement deux protocoles MAC. Cette utilisation alterna-

étoile.

2. La démarche de simulation est résumée à la fin du document, dans la partie 5.2.1.

tive de deux protocoles MAC peut être vue comme une architecture utilisant deux piles protocolaires. Dans la partie 3.2, nous généralisons cette approche d'architectures multi-piles. Nous définissons tout d'abord les architectures multi-piles simples, pour lesquelles les piles protocolaires sont indépendantes, dans le sens où chaque paquet est pris en charge par une seule pile de bout en bout. Nous définissons ensuite les architectures multi-piles locales, où chaque nœud peut décider localement d'utiliser une pile ou une autre pour la transmission des données. Dans la partie 3.3, nous montrons que ces architectures locales peuvent causer des boucles de routage, ce qui peut réduire les performances des protocoles. Nous donnons donc des critères sur les protocoles de routage afin de garantir l'absence de boucles. Plusieurs des résultats présentés dans ce chapitre sont issus des travaux avec Nassima HADID (pour la partie 3.1) et Nancy EL RACHKIDY (pour les parties 3.2 et 3.3), que j'ai co-encadrées pendant leur thèse, ainsi qu'avec trois étudiants en master recherche (pour la partie 3.1).

Dans le chapitre 4, nous partons d'un projet de surveillance de site environnemental, nommé Clervolc, pour approfondir certaines particularités des architectures multi-piles. Dans la partie 4.1, nous décrivons notre démarche initiale lors de ce projet, qui consiste à surveiller un volcan jusqu'à un évènement sismique critique, ce qui impose une longue phase d'attente (pendant laquelle les nœuds économisent leur énergie) puis une phase de travail intensif (pendant laquelle les nœuds doivent être très réactifs). Nous avons donc mené des réflexions autour de protocoles pour la phase d'attente et d'autres pour la phase de travail, ainsi qu'une architecture permettant de passer d'une phase à l'autre, suite à la détection d'un évènement sismique par l'un des nœuds. Dans la partie 4.2, nous généralisons cette approche en définissant les architectures multi-piles globales, pour lesquelles la pile protocolaire qu'un nœud doit utiliser est déterminée par un nœud distant. Nous proposons des mécanismes permettant de concevoir des protocoles adaptés à ces architectures globales. Dans la partie 4.3, nous étudions la manière dont le réseau évolue lorsqu'une architecture globale est utilisée, en modélisant cette évolution comme un graphe. Nous montrons que ce graphe d'évolution possède une structure de treillis, et nous utilisons les propriétés de cette structure pour obtenir des informations utiles pour le réseau (ou pour les utilisateurs du réseau), comme le temps nécessaire pour que tous les nœuds aient été notifiés de la nouvelle phase, ou l'ensemble des états possibles du réseau suite à l'obtention d'informations partielles sur l'état des nœuds. L'ensemble de ce chapitre est prospectif, et la plupart des résultats n'ont pas été validés par simulation ou maquettage. Les résultats initiaux de ce chapitre laissent donc entrevoir les perspectives des travaux présentés dans ce document. Ces résultats initiaux sont actuellement étudiés avec Affoua Thérèse ABY (pour les parties 4.1 et 4.2), que je co-encadre pendant sa thèse, et avec un étudiant en master recherche (pour la partie 4.1).

Finalement, dans le chapitre 5, nous faisons un résumé de nos contributions principales, ainsi que de quelques contributions non présentées. Certaines de ces contributions sont issues des travaux avec Nassima HADID et Nancy EL RACHKIDY. Puis, nous présentons de nombreuses perspectives de recherche à ces travaux, qu'il s'agisse de perspectives à court, moyen ou long terme.

CHAPITRE

2

ÉTAT DE L'ART

Un système de communication est généralement décomposé en services réseaux, qui sont fournis par des protocoles. Les services correspondant à des fonctionnalités proches sont regroupés en couches. Comme les services d'une couche s'appuient généralement sur les services de la couche inférieure, cette organisation en couches superposées est appelée une pile protocolaire.

Le modèle OSI (pour *Open Systems Interconnection*) [Zim80, OSI] est l'une des premières normes proposant une telle organisation. Ce modèle propose sept couches : la couche physique, la couche liaison de données, la couche réseau, la couche transport, la couche session, la couche présentation et la couche application. La couche physique gère la transmission de bits (*via* des signaux éventuellement codés et modulés). La couche liaison de données gère la communication entre machines voisines, et fournit un service d'adressage physique. Elle est souvent décomposée en deux sous-couches : la sous-couche MAC (*Medium Access Control*) et la sous-couche LLC (*Logical Link Control*). La sous-couche MAC gère l'accès au médium (en définissant quelles entités peuvent parler et quand). La sous-couche LLC gère habituellement les acquittements et les retransmissions de trames. La couche réseau gère les communications entre machines distantes, en fournissant les services d'adressage logique (permettant d'affecter une adresse à une machine en fonction de la topologie), de routage (permettant de calculer la route que doit suivre un paquet) et d'acheminement (permettant de faire suivre une route à un paquet). La couche transport gère les communications de bout en bout, en fournissant par exemple des services de garantie de livraison (incluant la retransmission de paquets, leur ordonnancement, et leur non-duplication), de contrôle de flux (permettant de ralentir un émetteur trop rapide) et de contrôle de congestion (permettant de réduire l'encombrement du réseau). La couche session gère l'ouverture et la fermeture de dialogues semi-permanents, en fournissant les services d'authentification et de reprise sur erreur.

La couche présentation gère le codage des données, en traduisant les données du format source au format réseau (éventuellement chiffré ou compressé), et du format réseau au format destination. Finalement, la couche application regroupe tous les programmes utilisant les ressources et services réseaux.

Comme nous l'avons dit précédemment, nous nous concentrons dans ce document sur la sous-couche MAC et la couche réseau du modèle OSI. Nous tirons aussi profit des avantages du *cross-layering*, c'est-à-dire des communications entre couches non adjacentes (par exemple *via* des échanges de paramètres spécifiques entre couches adjacentes).

Dans ce document, nous allons comparer les architectures traditionnelles, basées sur une unique pile protocolaire, avec les architectures constituées de plusieurs piles protocolaires.

Définition 1 (Architecture multi-piles). Une architecture est dite multi-piles quand plusieurs protocoles pouvant réaliser la même fonctionnalité coexistent dans cette architecture (et notamment, dans certains nœuds).

Il faut noter que cette définition peut présenter des ambiguïtés lorsqu'un protocole réalise une même fonctionnalité de différentes manières. On peut alors se poser la question du fait qu'il s'agisse d'un unique protocole ou non. En général, nous considérons qu'une architecture qui utilise un tel protocole est multi-piles.

Nous avons découpé cet état de l'art en deux parties. La partie 2.1 présente quelques architectures utilisant une unique pile protocolaire, nommées architectures mono-piles, et les protocoles associés (de la sous-couche MAC et de la couche réseau). La partie 2.2 présente des architectures utilisant plusieurs piles protocolaires, en incluant une justification de leur utilisation.

2.1 Architectures mono-piles et protocoles associés

Dans cette partie, nous nous concentrons sur les architectures mono-piles, c'est-à-dire les architectures pour lesquelles chaque fonctionnalité est prise en charge par un unique protocole (et fournie d'une seule manière).

L'objectif de cette partie de l'état de l'art n'est pas de donner une vision générale des protocoles de la sous-couche MAC et de la couche réseau pour les réseaux de capteurs sans fil (même si nous faisons référence à des articles présentant une vision générale de ces protocoles), mais simplement de rappeler que chaque protocole est conçu pour un type de trafic donné, et ne peut pas répondre à toutes les exigences de qualité de service. Pour les protocoles de la sous-couche MAC, nous présentons quelques protocoles basés sur la contention pour l'accès au médium, et d'autres basés sur un séquençement temporel des activités. Pour les protocoles de la couche réseau, nous décrivons uniquement les protocoles que nous utilisons dans la suite du document.

2.1.1 Protocoles de la sous-couche MAC

Il existe de nombreux protocoles MAC pour les réseaux de capteurs sans fil [NS04, DEA06, Cza06, ABJ08, YVM09, YIE11, HXS⁺13]. Dans la suite, nous décrivons quelques-uns des protocoles les plus classiques de la littérature, en les regroupant en trois catégories : les protocoles basés sur la contention, fonctionnant selon un algorithme de type CSMA (*Carrier-Sense Multiple Access*), les protocoles basés sur un séquençement temporel, fonctionnant selon un algorithme TDMA (*Time-Division Multiple Access*), et les protocoles hybrides, combinant des aspects CSMA et des aspects TDMA. Tous ces protocoles sont décrits très brièvement : notre objectif ici est seulement de rappeler qu'un même protocole MAC (quel qu'il soit) n'est pas adapté à tous les types de trafic.

2.1.1.1 Protocoles basés sur CSMA

Dans les protocoles basés sur CSMA, les tentatives d'accès au canal se font après une attente dont la durée est choisie aléatoirement dans un intervalle. Lorsque le canal est détecté occupé, ce qui fait échouer la tentative d'accès, une nouvelle tentative d'accès a lieu avec une attente choisie aléatoirement dans un intervalle plus grand. Cette augmentation de l'intervalle des valeurs d'attente réduit la probabilité de collision, mais réduit aussi les performances du mécanisme.

Dans le protocole S-MAC (*Sensor MAC*) [YHE04], les nœuds voisins s'organisent en groupes afin de définir un calendrier commun d'activité, pendant lequel les nœuds peuvent communiquer en CSMA, et de sommeil, pendant lequel les nœuds économisent de l'énergie. Lorsqu'un nœud appartient à plusieurs groupes, il doit se réveiller pendant l'activité de chacun de ses groupes. S-MAC utilise des messages RTS/CTS (*Request To Send / Clear To Send*) pour éviter les collisions lors de l'accès au médium. S-MAC permet généralement d'économiser de l'énergie en réduisant les durées d'écoute du médium.

Dans le protocole WiseMAC [EEHDP04], les transmissions de données sont précédées de longs préambules. Les récepteurs potentiels se réveillent périodiquement (de manière désynchronisée) et sont alertés de l'envoi de données à venir par le préambule. Les nœuds adaptent dynamiquement la taille du préambule en fonction de la connaissance des calendriers d'activation de leurs voisins. WiseMAC n'est pas adapté aux transmissions de messages de diffusion (pour lesquels plusieurs copies doivent être envoyées pour s'adapter aux différents calendriers d'activation des nœuds). De plus, WiseMAC souffre du problème du terminal caché.

De manière générale, les protocoles basés sur CSMA ne sont pas adaptés à des trafics importants, qui conduisent les nœuds à passer davantage de temps à résoudre la contention plutôt qu'à envoyer des données.

2.1.1.2 Protocoles basés sur TDMA

Dans les protocoles basés sur TDMA, le temps est divisé en intervalles, et chaque intervalle est attribué à un nœud donné. Un nœud n'est autorisé à transmettre que pendant l'intervalle de temps qui lui a été attribué, ce qui garantit un accès exclusif au canal et donc évite les collisions.

Dans le protocole DE-MAC [KRKI03], chaque nœud détermine les intervalles de temps de ses voisins, et choisit des intervalles (parmi ceux disponibles) pour ses propres transmissions. Le nombre d'intervalles de temps choisis par un nœud dépend de son énergie résiduelle. Les nœuds participent de plus à un mécanisme local d'élection, dans lequel ceux ayant le moins d'énergie résiduelle sont élus. Les perdants de l'élection réduisent leurs intervalles de temps, et les gagnants de l'élection augmentent leurs intervalles de temps : comme les nœuds propriétaires d'un intervalle de temps sont les seuls à pouvoir transmettre pendant cet intervalle (ou à dormir), les nœuds ayant le moins d'énergie peuvent économiser globalement plus d'énergie (en dormant davantage) que les autres (qui doivent écouter pendant les longs intervalles des autres nœuds). L'inconvénient majeur de DE-MAC est que les nœuds consomment beaucoup d'énergie en écoute du médium¹.

Dans le protocole S-TDMA [BBB12], des intervalles de temps sont assignés aux nœuds. Pour réduire le nombre de changements d'états du module radio, l'allocation des intervalles de temps vise à allouer des intervalles de temps consécutifs pour chaque nœud. S-TDMA est conçu pour s'attaquer au problème du délai de bout en bout, en intégrant les contraintes de délai dans l'allocation des intervalles. L'inconvénient majeur de S-TDMA est la surcharge qu'il impose quand le trafic est faible.

De manière générale, les protocoles basés sur TDMA ne sont pas adaptés à des trafics faibles. De plus, ils nécessitent des mécanismes complexes à mettre en place, qu'il s'agisse de mécanismes de synchronisation des nœuds (souvent basés sur un nœud maître) ou de mécanismes de dimensionnement des périodes (difficiles à adapter au trafic).

2.1.1.3 Protocoles hybrides

Dans le protocole TRAMA [ROGLA03], le temps est découpé en deux périodes : une période de contention, et une période sans contention. La période de contention utilise un mécanisme CSMA. La période sans contention utilise un mécanisme TDMA en divisant le temps en intervalles. Pendant chaque intervalle, un nœud parmi un voisinage à deux sauts est élu comme étant le seul émetteur autorisé à transmettre. Ainsi, TRAMA peut garantir qu'il n'y a pas de collisions dues à des terminaux cachés (dans la période TDMA), et peut garantir que les récepteurs sont éveillés lorsqu'un émetteur transmet. L'inconvénient principal de TRAMA est la consommation énergétique des nœuds pendant la période (obligatoire) de contention, ainsi que la consommation des ressources pour réaliser le calcul de l'allocation des intervalles de la période sans contention.

Dans le protocole Funneling-MAC [AMC⁺06], l'accumulation du trafic venant de l'ensemble des nœuds et à destination du puits est prise en compte en utilisant deux mécanismes : un mécanisme TDMA dans la zone proche du puits, et un mécanisme CSMA dans le reste du réseau. Les nœuds de la zone TDMA sont synchronisés au moyen d'une trame balise transmise par le puits avec une puissance élevée. Lors de

1. Il faut noter que les nœuds d'un réseau de capteurs sans fil consomment généralement davantage d'énergie en écoute (ou en réception) qu'en transmission [FvdBV12]. Par exemple, pour le composant CC2420 [CC2420], la consommation en écoute et en réception est de 19,7 mA, et la consommation en transmission varie entre 8,5 mA et 17,4 mA (selon la puissance de transmission).

la réception de cette balise, les nœuds passent en mode TDMA pendant une certaine durée, puis rebasculent sur CSMA (jusqu'à la prochaine balise). L'ordonnement des activités des nœuds dans le mode TDMA est calculé par le puits, qui liste les chemins empruntés par les paquets de données circulant dans la zone TDMA et en déduit un ordonnancement. De plus, le puits utilise la réutilisation spatiale pour réduire la durée de cet ordonnancement. L'avantage de Funneling-MAC est qu'il prend en compte le paradigme de communications *many-to-one*, souvent utilisé dans les applications des réseaux de capteurs sans fil. L'inconvénient principal de Funneling-MAC est qu'il suppose une génération de trafic de tous les nœuds du réseau, avec une congestion autour du puits seulement.

De manière générale, les protocoles hybrides arrivent à réaliser un compromis entre les protocoles basés sur CSMA et les protocoles basés sur TDMA. Toutefois, les protocoles hybrides souffrent généralement d'une baisse de performances quand le trafic est très élevé (puisque la période CSMA réduit les performances de la période TDMA) et quand le trafic est très faible (puisque la période TDMA, incluant la synchronisation, réduit les performances de la période CSMA).

2.1.2 Protocoles de la couche réseau

Il existe de nombreux protocoles de routage pour les réseaux de capteurs sans fil [AKK04, AY05, MBWP08, VOCA09, YM10, LZHL11]. Dans cette partie, nous ne décrivons que les protocoles que nous utilisons dans la suite de ce document.

Le protocole AODV (*Ad-hoc On demand Distance Vector routing*) [RFC 3561] est un protocole de routage réactif, c'est-à-dire établissant les routes à la demande, conçu pour les réseaux *ad hoc* (mais pouvant être appliqué dans les réseaux de capteurs sans fil). AODV est un protocole à vecteur de distances, c'est-à-dire que les nœuds s'échangent des informations condensées sur la topologie, en l'occurrence un coût pour atteindre chaque destination. Lorsqu'une source doit acheminer un paquet à une destination inconnue, cette source inonde le réseau avec une requête de route. Cette requête contient le coût courant du chemin depuis la source. Chaque nœud intermédiaire retransmet la requête de route en mettant à jour ce coût (si et seulement si le coût de la requête reçue est meilleur que le coût actuellement connu par le nœud intermédiaire). Lorsque la destination reçoit une requête de route d'un nœud intermédiaire, cette destination répond à ce nœud intermédiaire avec une réponse de route. Chaque nœud intermédiaire qui reçoit une réponse de route la retransmet au nœud duquel il avait reçu la meilleure requête de route, et établit une entrée d'acheminement pour la destination (contenant l'adresse du nœud d'où est venu la réponse). Lorsque la source reçoit des réponses de route, elle choisit la meilleure (en terme de coût), et commence à transmettre les données sur le chemin établi. Lorsqu'un lien entre deux nœuds devient inutilisable, AODV initie une procédure de réparation locale, en cherchant à passer par un autre nœud. De plus, AODV utilise des numéros de séquence de routes, ce qui lui permet d'éviter de traiter des informations périmées, et évite ainsi les risques de boucles. AODV a l'avantage de ne pas échanger de messages de contrôle pour les routes qui ne sont pas utilisées. Toutefois, la procédure d'inondation des requêtes de routes est longue et nécessite la transmission de nombreux messages de contrôle. AODV est adapté au trafic pé-

riodique (dont la période est inférieure à la durée de vie des routes), mais n'est pas adapté aux trafics d'alarmes sporadiques.

Le protocole OLSR (*Optimized Link State Routing protocol*) [RFC 3626] est un protocole de routage proactif, c'est-à-dire établissant les routes *a priori*, lui aussi conçu pour les réseaux *ad hoc*. OLSR est un protocole à états de liens, c'est-à-dire que les nœuds s'échangent leur connaissance de la topologie périodiquement. Plus spécifiquement, chaque nœud dans OLSR détermine des relais parmi ses voisins, appelés MPR. L'acheminement du trafic de contrôle se fait uniquement par l'intermédiaire de ces nœuds MPR (ce qui permet de réduire le nombre de transmissions nécessaires pour inonder le réseau). Les nœuds MPR ont aussi en charge de transmettre les informations concernant l'état de leurs liens dans tout le réseau, ce qui suffit à OLSR pour calculer des plus courts chemins vers des destinations quelconques. Finalement, les MPR sont utilisés pour acheminer les données d'un nœud à une destination. La sélection des MPR est soumise aux deux contraintes suivantes : l'ensemble des MPR d'un nœud n doit couvrir tous les voisins à deux sauts du nœud n (c'est-à-dire que tout nœud v qui est voisin à un ou deux sauts de n doit aussi être voisin de l'un des MPR de n), et un nœud v ne peut être MPR d'un nœud n que si v est un voisin symétrique de n (c'est-à-dire que n est aussi un voisin de v). OLSR a l'avantage de permettre aux nœuds non MPR d'économiser de l'énergie. OLSR n'est pas adapté aux réseaux de faible densité (où peu de nœuds sont MPR), de grande dimension ou très dynamiques (à cause de la surcharge nécessaire aux mises à jour de la topologie), et au trafic peu volumineux.

Le protocole de routage raccourci [KKP⁺07] est une amélioration d'un protocole de routage hiérarchique fonctionnant sur une topologie en arbre². Alors que le protocole de routage hiérarchique se limite à acheminer les paquets sur les liens de l'arbre, le protocole de routage raccourci peut, dans certains cas, transmettre des paquets aux voisins d'un nœud. Pour cela, le protocole de routage raccourci utilise une estimation de la distance à parcourir entre deux nœuds quelconques sur l'arbre. Ainsi, plutôt que de restreindre l'acheminement aux liens de l'arbre, le protocole de routage raccourci cherche parmi l'ensemble des voisins d'un nœud celui qui possède la plus petite distance estimée à la destination, et achemine les paquets à ce nœud. Le protocole de routage raccourci permet d'acheminer les données sans échange de messages de contrôle (excepté les messages de maintenance de l'arbre). Toutefois, ce protocole n'utilise pas les plus courts chemins pour acheminer les paquets, et peut avoir tendance à causer de la congestion sur certains liens, notamment ceux proches de la racine de l'arbre.

Dans [FB07] et [CMP⁺13], les auteurs étudient les mises à jour des tables de routage de protocoles à états de liens lorsqu'un changement de topologie se produit. Pendant la phase transitoire ayant lieu avant la convergence des tables de routage, il est possible que des boucles se produisent. Les auteurs proposent donc des séquences de mises à jour à appliquer pour garantir l'absence de boucles pendant ces phases transitoires, au détriment du délai de convergence. Cette problématique est proche de celle que nous étudions dans ce document, puisque nous cherchons aussi à résoudre des problèmes de boucles. Plus précisément, la problématique décrite dans

2. En l'occurrence, le protocole de routage raccourci est une amélioration du protocole de routage hiérarchique de ZigBee, décrit dans la partie 2.2.1.2.

ces travaux est complémentaire à la nôtre (si l'on se restreint aux protocoles à états de liens), pour les raisons suivantes.

- Les auteurs considèrent les décisions d'un unique protocole en présence d'un changement de topologie, alors que nous considérons les décisions de plusieurs protocoles en l'absence de changement de topologie.
- Les auteurs considèrent que le changement de topologie est lié à un unique routeur (généralement, lorsqu'il tombe en panne), alors que nous ne faisons pas d'hypothèses sur les décisions de routage.
- Nous faisons l'hypothèse que les protocoles de routage de chacune des piles protocolaires sont cohérents, et qu'ils ne créent donc pas de boucles par eux-mêmes. Cela revient à considérer que les mécanismes proposés dans [FB07] et [CMP⁺13] sont appliqués préalablement aux protocoles que nous utilisons.
- Notre approche est réactive (puisque'elle est centrée sur les paquets qui circulent) et avec un surcoût limité en calcul (puisque'il suffit d'évaluer deux fonctions localement, comme nous le verrons dans la partie 3.3), alors que les travaux précédents suivent une approche proactive (puisque'ils considèrent les tables de routage directement) et avec un surcoût plus important en calcul (puisque'il faut recalculer en chaque routeur la séquence de mises à jour).

2.1.3 Résumé des architectures mono-piles

Nous venons de décrire brièvement quelques protocoles de la sous-couche MAC et de la couche réseau. Lorsqu'une architecture est mono-pile, c'est-à-dire qu'elle ne se base que sur un protocole MAC et un protocole de routage, elle ne peut pas fournir une grande variété de qualités de service, même lorsque les protocoles sont optimisés. Un argument similaire est décrit dans [BHDK09], où les auteurs montrent qu'aucun protocole MAC basé sur des préambules n'est optimal, car chaque protocole atteint son comportement optimal uniquement pour une charge donnée.

2.2 Architectures multi-piles et protocoles associés

Dans cette partie, nous donnons quelques exemples d'architectures multi-piles. Nous détaillons d'abord les normes IEEE 802.15.4 et ZigBee, très utilisées dans les réseaux de capteurs sans fil, et nous montrons que ces normes mettent en place une architecture multi-piles. Ensuite, nous présentons d'autres exemples d'architectures multi-piles, certaines adaptées aux réseaux de capteurs sans fil, et d'autres non. Parmi ces exemples, nous présentons les architectures pour réseaux multi-usages, qui constituent une tendance récente des réseaux de capteurs sans fil, et qui constituent un socle propice aux architectures multi-piles. En effet, dans le contexte des réseaux multi-usages, la diversité des applications engendre une diversité des besoins de services réseaux, et il est difficile de disposer d'un unique protocole qui couvre ces besoins. Finalement, nous présentons les limites des architectures multi-piles.

2.2.1 Normes IEEE 802.15.4 et ZigBee

Les normes IEEE 802.15.4 et ZigBee constituent la base de nombreux réseaux de capteurs sans fil. Dans cette partie, nous décrivons ces deux normes.

2.2.1.1 Norme IEEE 802.15.4

La norme IEEE 802.15.4 [IEEE 802.15-4 v2011] décrit les couches basses d'un réseau sans fil personnel à basse consommation énergétique.

IEEE 802.15.4 définit deux types d'entités : les RFD (*Reduced Function Devices*) et les FFD (*Full Function Devices*). Les RFD sont des nœuds aux fonctionnalités réduites, très économes en énergie, et pouvant être connectés à des capteurs ou à des actionneurs. Les FFD sont des nœuds ayant l'ensemble des fonctionnalités décrites dans la norme, notamment la capacité à router ou à coordonner le réseau. Dans chaque réseau IEEE 802.15.4, l'un des FFD est le coordinateur du réseau. Le rôle du coordinateur est entre autres d'initier la formation de la topologie du réseau³ et d'attribuer des intervalles de temps garantis à certains nœuds.

IEEE 802.15.4 définit deux topologies : la topologie en étoile et la topologie pair à pair. Dans la topologie en étoile, tous les nœuds sont à portée du coordinateur, et communiquent par son intermédiaire. Dans la topologie pair à pair, beaucoup plus générale, les communications peuvent être multi-sauts, et la seule contrainte pour les chemins de communication est qu'ils ne passent que par des FFD. Toutefois, la formation d'une topologie pair à pair et la gestion du routage sortent du cadre de la norme (ces points sont détaillés dans la norme ZigBee).

Couche physique de IEEE 802.15.4.

La couche physique de IEEE 802.15.4 gère principalement l'activation et la désactivation du module radio, le test du médium, la transmission des trames et leur réception.

Couche liaison de données de IEEE 802.15.4.

La couche liaison de données de IEEE 802.15.4 gère principalement l'accès au médium, l'acquittement des trames, l'association des nœuds et la désassociation des nœuds. Le standard définit deux modes : le mode avec suivi de balises (lorsque le réseau nécessite une synchronisation ou de faibles latences) et le mode sans suivi de balises. Nous nous concentrons ici sur la description du mode avec suivi de balises, qui permet d'économiser davantage d'énergie.

Dans le mode avec suivi de balises, le temps est découpé selon une structure appelée supertrame. Les supertrames sont délimitées par des balises envoyées périodiquement par les FFD. La supertrame commence donc avec l'envoi ou la réception d'une balise, et est suivie d'une période active et d'une période inactive, comme indiqué sur la figure 2.1. La durée de la supertrame, notée BI , dépend du paramètre

3. Les primitives d'association et de désassociation des nœuds sont fournies par la sous-couche MAC de IEEE 802.15.4, et utilisées par les couches supérieures (voir la partie 5.1.3 de [IEEE 802.15-4 v2011]).

$BO \in [0; 14]$ présent dans chaque balise. Plus précisément, on a $BI = 15,36 \cdot 2^{BO}$ ms. La durée de la période active, notée $SD \in [0; BI]$, dépend du paramètre SO aussi présent dans chaque balise. Plus précisément, on a $SD = 15,36 \cdot 2^{SO}$ ms. Pendant la période inactive (de durée $BI - SD$), le nœud peut économiser de l'énergie en désactivant son module radio.

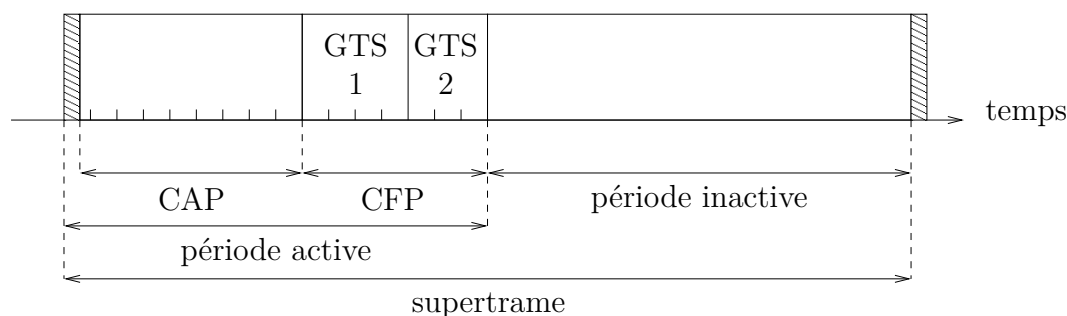


Figure 2.1 – Un exemple de supertrame de IEEE 802.15.4 en mode avec suivi de balises. Les balises sont représentées en hachuré.

La période active est découpée en seize intervalles identiques, regroupés en deux périodes : une CAP (*Contention Access Period*) et une CFP (*Collision-Free Period*) optionnelle. La durée de la CFP est égale au nombre d'intervalles réservés dans la balise (sur l'exemple de la figure, il y a sept intervalles réservés). La CAP occupe la partie restante de la période active (sur l'exemple de la figure, il y a neuf intervalles pour la CAP, moins la durée d'envoi ou de réception de la balise).

Durant la CAP, l'accès au médium est fait selon l'algorithme CSMA/CA slotté. Cet algorithme utilise pour base de temps un intervalle de $320 \mu\text{s}$: chaque opération commence à la frontière de l'un de ces intervalles. Pour transmettre une trame, chaque nœud commence par attendre un nombre aléatoire d'intervalles, tiré dans $[0; 2^{BE} - 1]$, BE étant initialement égal à 3. Puis, le nœud effectue deux tests de médium, appelés CCA (*Clear Channel Assessment*), afin de déterminer si une transmission d'un autre nœud est en cours. Si l'un des CCA indique que le médium est occupé, le nœud incrémente BE (tout en ne dépassant pas la valeur 5), et recommence l'attente de durée aléatoire. Cette répétition est effectuée au plus quatre fois : si le médium n'a pas pu être trouvé libre après ce nombre de répétitions, l'algorithme indique un échec d'accès au médium. Si les deux CCA indiquent que le médium est libre, l'algorithme indique un succès pour l'accès au médium. Lorsque la tentative d'accès au médium est faite avec succès, le nœud envoie la trame immédiatement, et attend la réception d'un acquittement. Si aucun acquittement n'est reçu, ou si la tentative d'accès au médium a échoué, le nœud renouvelle la tentative d'accès au médium. Le nœud effectue un maximum de quatre tentatives d'accès au médium avant de considérer que la trame n'a pas pu être envoyée (et il en informe alors la couche supérieure).

Durant la CFP, l'accès au médium est fait au moyen d'intervalles de temps garantis, appelés GTS (*Guaranteed Time Slots*). Ces intervalles sont négociés durant la CAP des supertrames précédentes. Seul le nœud propriétaire d'un GTS est autorisé à transmettre pendant ce GTS. Durant un GTS, l'accès au médium est donc fait de manière directe.

La norme IEEE 802.15.4 propose donc deux mécanismes MAC différents, utilisés conjointement (mais à des instants différents : l'un des mécanismes est utilisé durant la CAP, et l'autre durant la CFP). La norme IEEE 802.15.4 constitue donc une architecture multi-piles.

2.2.1.2 Norme ZigBee

La norme ZigBee [ZigBee] décrit les couches hautes d'un réseau sans fil personnel à basse consommation énergétique, en s'appuyant sur la couche physique et la sous-couche MAC de IEEE 802.15.4 [IEEE 802.15-4 v2003] pour les couches basses. Dans cette partie, nous nous concentrons sur la description de la couche réseau de ZigBee.

ZigBee définit trois types d'entités : les feuilles, les routeurs et le coordinateur. Les feuilles sont des nœuds ayant des fonctionnalités limitées, et sont généralement équipés de capteurs ou d'actionneurs. Il peut s'agir de RFD ou de FFD de IEEE 802.15.4. Les routeurs sont des nœuds pouvant participer à l'acheminement des données, ainsi qu'à certaines tâches de maintenance de la topologie. Il s'agit obligatoirement de FFD de IEEE 802.15.4. Le coordinateur est un nœud crucial dans ZigBee, puisqu'il est responsable de la création du réseau et du choix de certains paramètres. Il s'agit du coordinateur de IEEE 802.15.4.

ZigBee définit trois topologies : l'étoile, l'arbre, et la topologie maillée. Dans la topologie en étoile, le réseau est contrôlé par le coordinateur, par lequel toutes les communications passent. Dans les topologies en arbre et maillées, le coordinateur initie la construction du réseau et détermine les paramètres globaux, mais la topologie peut être étendue par l'intermédiaire de routeurs. Dans la suite, nous décrivons uniquement les topologies en arbre et maillées.

Topologie en arbre.

Dans la topologie en arbre, la sous-couche MAC utilisée est soit IEEE 802.15.4 en mode sans suivi de balises, soit une version modifiée de IEEE 802.15.4 en mode suivi de balises (voir les parties 1.1.4 et 3.6.4 de [ZigBee]). Nous détaillons ici cette version modifiée de IEEE 802.15.4 en mode suivi de balises, puisqu'il s'agit d'une sous-couche MAC permettant d'économiser de l'énergie. Puis, nous présentons l'adressage hiérarchique et le routage hiérarchique de ZigBee.

Dans la version modifiée de IEEE 802.15.4 en mode suivi de balises, ZigBee profite du fait que le taux d'activité des nœuds est généralement très faible (pour que les nœuds puissent économiser de l'énergie) pour placer les activités des nœuds d'un même voisinage sur des intervalles de temps disjoints. Dans sa balise, chaque routeur indique l'instant de transmission de son père, afin de réduire le problème du terminal caché. Lorsqu'un nouveau routeur rejoint le réseau, soit il identifie (après avoir écouté le médium suffisamment longtemps) une période disponible pour sa période d'activité, soit il agit comme une feuille. Chaque nœud gère son temps à partir des balises de son père afin de corriger les dérives éventuelles.

ZigBee propose un mécanisme hiérarchique pour l'allocation des adresses (conçu pour la topologie en arbre) implémenté par les routeurs et par le coordinateur. Cet adressage fonctionne de la manière suivante. Le coordinateur du réseau utilise trois

paramètres (qui sont transmis à tous les nœuds du réseau) : le nombre maximum C_m de fils par routeur (ou pour le coordinateur), le nombre maximum R_m de fils routeurs par routeur, et la profondeur maximale L_m de la topologie. Le coordinateur s'attribue l'adresse 0. Chaque routeur n détermine, en fonction de sa profondeur d_n dans l'arbre, le nombre maximum $Cskip(d_n - 1)$ de nœuds qu'il peut avoir dans sa descendance (lui inclus). n en déduit ainsi sa plage d'adresses, s'affecte la première adresse de cette plage d'adresses, répartit pour chacun de ses R_m routeurs fils potentiels une plage d'adresses de taille $Cskip(d_n)$, puis attribue les adresses restantes à ses feuilles filles.

Le routage hiérarchique de ZigBee se base sur l'adressage hiérarchique. Pour transmettre une trame t à une destination d'adresse d , un nœud n applique l'algorithme suivant. Si n est une feuille, le prochain saut de t est le père de n dans l'arbre. Sinon, n détermine si la destination est dans sa descendance, en vérifiant si $n < d < n + Cskip(d_n - 1)$. Si d n'est pas dans la descendance de n , le prochain saut de t est le père de n dans l'arbre. Si d est dans la descendance de n , n détermine si d correspond à une feuille fille ou non, en vérifiant si $d < n + R_m \cdot Cskip(d_n) + C_m$. Si d correspond à une feuille fille de n , le prochain saut de t est d . Sinon, le prochain saut de t est le routeur fils de n dont l'adresse est $n + 1 + \lfloor (d - (n + 1)) / Cskip(d_n) \rfloor \cdot Cskip(d_n)$.

Topologie maillée.

Dans la topologie maillée, la sous-couche MAC utilisée est IEEE 802.15.4 en mode sans suivi de balises (voir les parties 1.1.4 et 3.6.4 de [ZigBee]), bien que cette sous-couche MAC ne permette pas nativement d'économiser de l'énergie. Les adresses attribuées dans cette topologie sont généralement aléatoires : lorsqu'un conflit d'adresses est détecté, une procédure de changement d'adresses est lancée.

Le protocole de routage de ZigBee cherche à construire des chemins de coût minimum. Lorsque plusieurs chemins de coût minimum existent, ZigBee adopte le plus court d'entre eux (en terme de nombre de sauts). Le coût d'un chemin est défini comme la somme du coût des liens du chemin. Le coût d'un lien l_i est défini par $\min\{7, \lfloor p_i^{-4} \rfloor\}$, où p_i est la probabilité de livraison d'un paquet sur ce lien l_i .

Le protocole de routage de ZigBee est très similaire à AODV.

2.2.2 Autres architectures multi-piles

Dans cette partie, nous donnons des exemples d'architectures multi-piles et de protocoles pouvant être considérés comme appartenant à des architectures multi-piles.

2.2.2.1 Pile duale IPv4-IPv6

L'architecture multi-piles la plus connue est probablement la pile duale IPv4-IPv6 [RFC 4213], qui est une implémentation du protocole IPv6 permettant de simplifier la cohabitation avec le protocole IPv4. Dans la pile duale IPv4-IPv6, la pile protocolaire IPv4 et la pile protocolaire IPv6 cohabitent, ce qui permet aux nœuds IPv6 implémentant la pile duale IPv4-IPv6 de pouvoir communiquer avec les nœuds ne comprenant que IPv4. De tels nœuds IPv6 peuvent communiquer avec des nœuds

IPv4 *via* IPv4, et avec des nœuds IPv6 *via* IPv6. De plus, l'infrastructure de réseau IPv4 peut être utilisée pour acheminer les paquets IPv6 en utilisant une technique de tunnels aux frontières des réseaux qui ne comprennent que IPv4 : à l'entrée d'un tunnel, les paquets IPv6 sont encapsulés dans des paquets IPv4, et à la sortie du tunnel, les paquets IPv6 sont désencapsulés. L'utilisation de la pile IPv4 ou de la pile IPv6 est déterminée par l'application [RFC 3493]. Comme les structures de données internes correspondant aux *sockets* IPv4 et aux *sockets* IPv6 ne sont généralement pas compatibles, le choix est souvent imposé de manière logicielle lors de l'implémentation de l'application. On peut noter que les interactions entre ces deux piles sont *a priori* très limitées, à l'exception de l'utilisation d'adresses IPv6 correspondant à des adresses IPv4. Une adresse IPv6 correspondant à une adresse IPv4 est une adresse écrite au format IPv6 (sur 128 bits) permettant de représenter une adresse IPv4 (sur 32 bits). Ces adresses permettent à une application IPv6 pure de dialoguer avec une application IPv4 (éventuellement distante).

2.2.2.2 Protocoles MAC pour architectures multi-piles

Nous décrivons ici quelques protocoles MAC agissant différemment en fonction du marquage des trames (généralement, choisi par la couche applicative en fonction de la priorité des trames). Une architecture intégrant l'un de ces protocoles MAC peut être considérée comme multi-piles.

La norme IEEE 802.11e [IEEE 802.11e v2005] intègre plusieurs types d'accès au canal [DLPF09] : le DCF (*Distributed Coordination Function*), le PCF (*Point Coordination Function*), le HCF (*Hybrid Coordination Function*) avec accès à contention, et le HCF avec accès contrôlé. Le DCF utilise un accès basé sur l'algorithme CSMA/CA (*CSMA with Collision Avoidance*). Le PCF consiste à interroger les stations à tour de rôle. Le HCF avec accès à contention utilise aussi l'algorithme CSMA/CA, combiné avec le mode EDCA (*Enhanced Distributed Channel Access*) pour prioriser le trafic en fonction de son type, le mode EDCA ayant un impact notablement sur la taille des fenêtres de tirage de retards. Le HCF avec accès contrôlé utilise le mode HCCA (*HCF Controlled Channel Access*), qui généralise le PCF en permettant de découper le temps dynamiquement en plusieurs intervalles. Ces types d'accès au canal ne sont pas utilisés conjointement, mais dans des intervalles de temps différents (voir la partie 9.1.4 de [IEEE 802.11e v2005]).

Le protocole PQ-MAC [PKSH07] découpe le temps en une phase de configuration et une phase d'émission de données, qui se répètent de manière périodique. Lors de la phase de configuration, les nœuds se partagent des intervalles de temps en fonction de la priorité des trames qu'ils ont à transmettre. Lors de la phase d'émission de données, le nœud propriétaire d'un intervalle de temps peut envoyer ses trames prioritaires. Si le nœud propriétaire n'a aucun trame prioritaire à envoyer, les nœuds non prioritaires peuvent entrer en compétition pour envoyer leurs trames. Ce protocole utilise donc une approche TDMA pour les trames prioritaires et une approche CSMA pour les autres trames.

Le protocole I-MAC [SSJZ08] utilise un mécanisme similaire pour l'accès au médium. Dans ce protocole, le temps est découpé en intervalles de temps qui sont alloués aux nœuds en fonction de la priorité des trames qu'ils doivent transmettre.

Le nœud propriétaire d'un intervalle de temps a un accès garanti à cet intervalle, et peut transmettre ses données dès le début de l'intervalle. Lorsqu'un nœud prioritaire n'a plus de données à transmettre, ou lorsqu'un intervalle de temps n'a aucun propriétaire, les nœuds non propriétaires peuvent entrer en compétition pour l'accès au médium. La différenciation de services entre les nœuds non propriétaires est faite en faisant varier la taille des fenêtres de contention.

2.2.2.3 Protocoles de routage pour architectures multi-piles

Nous décrivons ici quelques protocoles de routage agissant différemment en fonction du marquage des paquets (généralement, choisi par la couche applicative en fonction de la priorité des paquets). Une architecture intégrant l'un de ces protocoles de routage peut être considérée comme multi-piles.

Le protocole de routage RGR [SSHK⁺12] combine un protocole réactif (en l'occurrence, AODV) avec du routage géographique glouton. Lorsqu'il n'existe pas de route pour une destination donnée, le nœud source initie le processus de découverte de route avec AODV en inondant le réseau de requêtes de route, et en attendant les réponses de route. Les réponses de route de RGR incluent des informations de localisation. Une fois que ces routes sont établies par AODV, elles sont utilisées pour l'acheminement des paquets. Lorsqu'un lien est considéré inutilisable, RGR bascule en mode géographique glouton (pour la destination concernée). Dans le mode géographique glouton, les paquets sont acheminés de proche en proche, en tentant à chaque saut de s'approcher le plus possible de la destination.

Dans [MHJ⁺12], les auteurs proposent une stratégie de routage nommée R_{e+d} qui combine une stratégie R_d visant à minimiser le délai avec une stratégie R_e visant à minimiser l'énergie utilisée. Les paquets considérés urgents (par l'application) sont acheminés avec la stratégie de routage R_d , et les autres paquets sont acheminés avec la stratégie de routage R_e . Les auteurs montrent que la stratégie R_{e+d} réalise un compromis entre délai et énergie.

Dans [FCA10], les auteurs proposent un protocole de routage pour gérer deux types de paquets dans un réseau de capteurs sans fil : des paquets d'alertes, de haute priorité, et des paquets périodiques, de basse priorité. Les paquets sont placés dans des files d'attente différentes en fonction de leur priorité, et sont traités par ordre de priorité décroissante. Le choix du prochain saut pour un paquet donné est basé sur une fonction de coût dont la pondération dépend de la priorité du paquet, et prenant en considération la distance à la destination, la qualité du lien vers le prochain saut, l'énergie résiduelle du prochain saut, et la charge du prochain saut.

2.2.2.4 Réseaux de capteurs sans fil multi-usages

Les réseaux de capteurs sans fil ayant des capacités très limitées, plusieurs chercheurs se sont concentrés sur des optimisations fines concernant le matériel, le logiciel ou les protocoles. Ces optimisations sont souvent spécifiques à l'application déployée. Par exemple, les auteurs de [LGC04] proposent une optimisation logicielle qui fonctionne de la manière suivante. Dans un premier temps, le développeur programme son application pour qu'elle soit exécutée sur une machine virtuelle (plutôt que sur la plateforme matérielle directement). Puis, un logiciel d'optimisation propose un

jeu d'instructions pour la machine virtuelle, qui est spécifique à l'application développée. Ainsi, en fonction de la fréquence d'utilisation des instructions, le code déployé sur le nœud capteur peut avoir une taille réduite. Cet exemple, datant de 2004 confirme le fait qu'il y a quelques années, un réseau de capteurs sans fil était généralement conçu pour une seule application.

Un réseau de capteurs sans fil multi-usages [SFCB05] est un réseau de capteurs déployé pour supporter plusieurs applications, afin de mutualiser le coût du déploiement.

L'idée des réseaux de capteurs sans fil multi-usages semble avoir émergé vers 2005. Les travaux présentés dans [SFCB05] mettent en avant les avantages économiques des réseaux de capteurs sans fil multi-usages. Les auteurs montrent que si le réseau de capteurs peut être vu comme un moyen de fournir un service modulaire et flexible à plusieurs applications, son utilité augmente et son coût total est réduit. De même, les travaux présentés dans [SZLD12] mettent en avant les avantages de disposer de réseaux de capteurs sans fil sans limitation énergétique, permettant l'émergence de nouveaux services reposant sur une architecture flexible et passant à l'échelle. Finalement, les travaux présentés dans [WMG04] considèrent que le réseau de capteurs sans fil peut être accédé par l'intermédiaire d'un intergiciel, qui se traduit (pour ces auteurs) par une base de données traditionnelle pouvant être interrogée par un langage de type SQL. Cette approche revient à faire disparaître les spécificités liées à l'application, et revient à abstraire jusqu'à la notion de réseau. En découplant ainsi l'application des protocoles réseaux, il est possible d'envisager plusieurs applications pour un même déploiement, de manière complètement transparente pour le réseau (puisque les applications n'interagissent avec le réseau qu'au travers des requêtes transmises).

Dans la suite, nous décrivons des travaux de la littérature basés sur deux types de réseaux de capteurs sans fil multi-usages : les réseaux de capteurs sans fil conçus pour des applications génériques, et les réseaux de capteurs sans fil virtuels.

Réseaux de capteurs sans fil pour applications génériques.

Plusieurs travaux sur les réseaux de capteurs sans fil se basent sur des applications très génériques, aux besoins variés (voire inconnus *a priori*). Parmi les travaux cités, certains mettent en avant explicitement le fait que le réseau de capteurs est multi-usages, dans le sens où plusieurs applications cohabitent, alors que d'autres considèrent implicitement une unique application. Toutefois, lorsque l'application offre des fonctionnalités très variées, voire inconnues au moment du déploiement, nous considérons qu'il peut s'agir de plusieurs applications différentes.

Les travaux de [SFCB05] traitent de réseaux de capteurs sans fil pour le transport de marchandises. Les capteurs peuvent être utilisés pour surveiller la température, dans le cas de marchandises périssables. Les capteurs peuvent aussi être utilisés pour faire du suivi de conteneurs, ou pour localiser les conteneurs. Chaque application n'interroge qu'un sous-ensemble des capteurs, en fonction du besoin. Des extensions de sécurité sont prévues dans [SV05] dans le cas où les entités concernées par chaque application (par exemple, les propriétaires des conteneurs, les transporteurs, ou les autorités) ne doivent pas toutes avoir accès aux mêmes informations.

Les travaux de [MG07] présentent une application domotique, et donnent divers besoins. Parmi ces besoins, les auteurs décrivent la détection d'infractions, le suivi des consommations d'électricité et d'eau, la surveillance de personnes âgées ou d'enfants, et l'évolution de la pollution de l'air. Les auteurs proposent un intergiciel permettant de répondre aux problèmes de sélection des nœuds à utiliser (en fonction des besoins applicatifs), de sécurité (pour lutter contre les problématiques liées à la capture de nœuds), et d'identification des nœuds.

Les travaux de [JHMJ10, JMJD10, Jav11] présentent des réseaux de capteurs sans fil pouvant fournir des services à plusieurs applications concurrentes. L'application visée est la surveillance de plusieurs paramètres dans un entrepôt (afin de surveiller la température, la lumière, la position des camions, et la manutention des colis). Les auteurs se concentrent sur deux aspects : la manière dont la qualité des données peut être garantie (en faisant de la réservation de ressources et de la sélection automatique de fournisseur de service) [JHMJ10, Jav11], et le partage de composants logiciels afin d'éviter les composants ayant des fonctionnalités redondantes [JMJD10].

Enfin, quelques travaux décrivent des algorithmes pour des applications génériques, sans détailler l'application considérée. Par exemple, les travaux de [TOOM08] proposent une solution de type multi-agents pour gérer les ressources énergétiques des nœuds, et pour allouer les ressources dans un environnement dynamique (étant donné que les ressources des nœuds qui dorment ne peuvent pas être utilisées). Les travaux de [WVG04, MFHH05] présentent des mécanismes permettant d'interroger un réseau de capteurs comme s'il s'agissait d'une base de données générique, avec un langage proche du SQL.

Réseaux de capteurs sans fil virtuels.

Un réseau de capteurs sans fil virtuel est un réseau collaboratif dans lequel les ressources des nœuds sont partagées par plusieurs applications. L'objectif de ces réseaux est de profiter de la collaboration des différents nœuds pour répondre aux besoins de l'application d'une manière efficace en ressources.

Les premiers réseaux de capteurs sans fil virtuels proposés [LXX⁺06, SHC⁺08] se concentraient sur l'interconnexion de réseaux différents et éloignés géographiquement, au moyen d'une interface uniforme. La collaboration des réseaux se limitait alors au partage de données pour des applications nécessitant des sources d'informations diverses.

Puis, les réseaux de capteurs sans fil virtuels ont pris une autre dimension [JHI07, BJI08, SZLD12], en évoluant vers une infrastructure légère pour laquelle les ressources (notamment en termes de calcul et de communication) sont supérieures aux besoins d'une seule application. Ainsi, les travaux de [JHI07] montrent que les réseaux de capteurs sans fil virtuels peuvent permettre de collecter des données issues de réseaux qui se superposent géographiquement, en prenant comme exemple le suivi de la contamination d'une nappe d'eau souterraine. Les travaux de [BJI08] présentent des techniques d'optimisation réseau pour les réseaux de capteurs sans fil multi-usages, afin d'améliorer la performance des communications. Les travaux de [SZLD12] se concentrent sur les mécanismes permettant de rendre des réseaux

de capteurs sans fil interopérables, et capables de fournir les services attendus. Les travaux de [TLZ⁺13] envisagent que les réseaux de capteurs sans fil virtuels peuvent être interconnectés avec Internet, et proposent de nombreuses applications allant des maisons intelligentes aux villes intelligentes.

2.2.3 Limites des architectures multi-piles

Les architectures multi-piles souffrent de plusieurs limites, comme l'importance de la taille du code (puisque chaque protocole de chaque pile doit être implémenté), l'importance du nombre de données à manipuler (puisque chaque protocole de chaque pile a besoin de données), la difficulté de dimensionnement des périodes associées à chaque pile, ou les problèmes d'interopérabilité entre les protocoles des différentes piles.

Les problématiques de taille du code ou des données sont souvent au cœur des critiques des architectures multi-piles. Les nœuds capteurs ayant généralement des capacités de stockage très limitées (par exemple, les nœuds TelosB [Mem13] disposent de 48 ko de mémoire pour le code et de 10 ko de mémoire pour les données), il peut être contraignant d'implémenter plusieurs protocoles. Toutefois, ces limites peuvent être atténuées en considérant les éléments suivants. Tout d'abord, une partie du code (et, dans une moindre mesure, des données) peut être partagée par les protocoles dans les architectures multi-piles. C'est d'ailleurs cette considération qui est au centre de la proposition de [JMJD10]. Ensuite, la norme principale pour les réseaux de capteurs sans fil, à savoir IEEE 802.15.4, est par nature multi-piles. Les protocoles proposés dans ces normes (pour au moins l'une des piles) sont suffisamment simples pour que la surcharge en taille de code soit réduite.

Les problèmes de dimensionnement des périodes associées à chaque pile sont généralement difficiles à résoudre. La solution adoptée par la plupart des architectures est de laisser à l'utilisateur le soin de paramétrer la durée de chaque période (de manière absolue ou relative par rapport à la durée totale d'un cycle). Les mécanismes que nous proposons dans 3.2.2.1 permettent de réduire les problèmes de dimensionnement des périodes, puisque le trafic peut être acheminé dans la plupart des périodes.

Les problèmes d'interopérabilité entre les protocoles de différentes piles sont généralement évités en utilisant l'une des deux approches suivantes. La première approche consiste à se limiter à utiliser des protocoles interopérables. Par exemple, lorsque l'architecture multi-piles utilise des protocoles MAC différents, ces protocoles sont souvent conçus pour être intégrés dans un même mécanisme (par exemple, c'est le cas pour IEEE 802.15.4 avec la période obligatoire CAP et la période optionnelle CFP). Lorsque l'architecture multi-piles utilise des protocoles réseaux différents, ces protocoles sont conçus pour être interopérables (par exemple, c'est le cas pour la pile duale IPv4-IPv6). L'inconvénient de cette approche est que la conception des protocoles devient ardue, et qu'il n'est pas possible d'utiliser dans chaque pile des protocoles arbitraires. La deuxième approche consiste à imposer que chaque paquet de données soit pris en charge par une seule pile protocolaire (ce qui correspond à la définition présentée dans la partie 3.2.1). La suppression de cette hypothèse de travail simplificatrice est étudiée dans la partie 3.2.2.

2.2.4 Résumé des architectures multi-piles

Nous venons de voir que plusieurs architectures réseaux existantes peuvent être considérées comme des architectures multi-piles, dans le sens où elles exploitent plusieurs protocoles de la sous-couche MAC ou de la couche réseau pour fournir plusieurs qualités de service. Nous avons cité, notamment, les normes IEEE 802.15.4 et ZigBee, la pile duale IPv4-IPv6, ou encore IEEE 802.11e. Un autre type d'architecture multi-piles concerne les réseaux de capteurs sans fil qui ne sont pas prévus pour une application spécifique, mais pour des applications génériques.

2.3 Bilan

De nombreux travaux de recherche visent à optimiser les performances de protocoles de la sous-couche MAC ou de la couche réseau dans des conditions spécifiques (c'est d'ailleurs le cas dans plusieurs de nos travaux, comme par exemple [ERGB13*, ERGM10*, HGM09b*]). De manière générale, les gains obtenus en optimisant les protocoles peuvent être importants, et le champ d'application de ces protocoles peut être légèrement augmenté (pour qu'un protocole fonctionnant dans un contexte spécifique puisse être utilisé dans un contexte plus général) ou spécialisé (pour qu'un protocole fonctionnant dans un contexte général puisse être utilisé efficacement dans un contexte spécifique).

Toutefois, l'optimisation de protocoles de la sous-couche MAC et de la couche réseau ne permet pas de rendre les protocoles performants dans tous les contextes : l'optimisation n'est pas toujours suffisante.

Pour répondre à des besoins de qualité de service variés, la solution consistant à utiliser plusieurs protocoles pour une même fonctionnalité semble prometteuse (cette direction est d'ailleurs prise dans la plupart des normes récentes). Nous avons été amené, par l'intermédiaire des projets auxquels nous avons participé, à travailler sur les architectures multi-piles pour fournir des qualités de service variées. Dans les thèses de Nassima HADID et de Nancy EL RACHKIDY, que j'ai co-encadrées, et dans la thèse d'Affoua Thérèse ABY, que je co-encadre actuellement, la notion d'architectures multi-piles n'était pas initialement formalisée, mais la multiplicité des protocoles était essentielle. Ces travaux nous ont conduit à nous concentrer sur ces architectures multi-piles, qui constituent le cœur de ce document.

CHAPITRE

3

DÉCISION LOCALE DE CHANGEMENT DE PILES PROTOCOLAIRES

De nombreuses applications pour les réseaux de capteurs sans fil produisent plusieurs types de trafic : du trafic de surveillance périodique pour suivre l'évolution lente d'un phénomène, du trafic urgent lors de la détection d'évènements importants, du trafic de contrôle pour la gestion du réseau, etc. Chaque type de trafic à ses propres caractéristiques, et nécessite une certaine qualité de service. La tendance des réseaux de capteurs sans fil multi-usages vient accentuer ce constat : dans de tels réseaux, ce n'est plus la durée de vie qui est l'enjeu principal, mais le nombre d'applications qui peuvent partager le même déploiement [SFCB05] ; plusieurs applications opèrent sur le même réseau de capteurs sans fil, et le nombre de types de trafic que le réseau doit pouvoir gérer augmente significativement.

Il est très difficile de répondre aux multiples exigences de qualité de service, notamment lorsqu'elles sont divergentes, avec une unique pile protocolaire (même optimisée). La solution étudiée ici consiste à faire cohabiter plusieurs piles protocolaires, chacune prenant en charge un certain nombre de types de trafic.

Dans la partie 3.1, nous introduisons le projet Ocari, dont l'objectif est de concevoir un ensemble de protocoles pour les réseaux de capteurs sans fil industriels. L'une des spécificités à l'origine de ce projet est d'introduire une différenciation de services au niveau de la couche liaison de données. Dans la proposition formulée, les impacts de cette différenciation de services se retrouvent au niveau de la couche réseau et de la couche applicative. Ces impacts mettent en avant la cohabitation de deux piles protocolaires. Dans la partie 3.2, nous généralisons cette approche en proposant deux définitions d'architectures dans lesquelles plusieurs piles protocolaires peuvent

coexister : les architectures multi-piles simples, où un paquet est toujours géré par la même pile protocolaire, et les architectures multi-piles locales, où chaque nœud décide localement de la pile protocolaire la plus adaptée pour un paquet. Nous avons montré pendant la thèse de Nancy EL RACHKIDY que dans certaines conditions, des boucles de routage peuvent apparaître dans les architectures multi-piles locales. Dans la partie 3.3, nous proposons des solutions pour éviter ces boucles (ou pour réduire leurs effets).

3.1 Étude d'un cas concret : le projet Ocari

Le projet Ocari (Optimisations des Communications Ad-hoc pour les Réseaux Industriels) consiste à spécifier et à développer une pile protocolaire pour les réseaux de capteurs sans fil industriels, à très faible mobilité. Le projet a démarré avec un financement de l'ANR (le 20 mars 2007), et regroupait alors sept partenaires : quatre partenaires académiques (l'ex ICARE (devenu l'ex LATTIS puis l'IRIT), Inria, le LIMOS et le LRI) et trois partenaires industriels (la DCNS, EDF R&D et l'ex One RF (devenu Telit)). Entre la fin du financement ANR (en fin 2010) et début 2012, le projet Ocari a continué grâce au financement d'EDF R&D.

Les partenaires du projet se sont organisés de la manière suivante : la DCNS et EDF R&D ont fourni les besoins applicatifs et ont conçu les applications au niveau du puits de données, Inria a conçu les protocoles de la couche réseau, One RF a réalisé une plateforme matérielle et a implémenté les protocoles de la couche réseau, le LRI a modélisé la consommation énergétique des nœuds, et le LATTIS et le LIMOS ont conçu la sous-couche MAC et ont implémenté les protocoles MAC.

3.1.1 Objectifs

Les objectifs principaux du projet, à l'origine de la pile protocolaire Ocari, sont :

- le déterminisme au niveau de la sous-couche MAC pour certaines trames, c'est-à-dire le fait que pour ces trames, les pertes ne soient liées qu'au médium et non pas à la méthode d'accès,
- le délai de bout en bout borné pour certaines trames,
- l'économie d'énergie pour tous les nœuds.

Pour réaliser ces objectifs, les partenaires ont décidé du contexte suivant :

- la couche physique retenue est celle du standard IEEE 802.15.4,
- l'économie d'énergie doit se faire dès la couche physique et à tous les niveaux,
- deux types de trafic applicatif sont envisagés : le trafic contraint en temps, et le trafic non contraint en temps,
- les applications doivent être compatibles, dans la mesure du possible, avec les profils applicatifs de ZigBee et avec les applications métiers d'EDF R&D et de la DCNS.

Il a aussi été décidé d'utiliser avantageusement le *cross-layering* pour améliorer les performances des protocoles.

La pile protocolaire Ocari [DD08] est représentée sur la figure 3.1. La couche physique est celle du standard IEEE 802.15.4, sans modification. La couche liaison de

données est gérée par le protocole Macari (voir la partie 3.1.2.1) pour les aspects temporels et l'accès au médium (en CSMA/CA slotté ou en TDMA), et par le protocole Serena (voir la partie 3.1.2.2) pour l'économie d'énergie liée au trafic non contraint en temps. La couche réseau est gérée par le protocole Rocari (voir la partie 3.1.3.1) pour le trafic contraint en temps, et par le protocole EOLSR (voir la partie 3.1.3.2) pour le trafic non contraint en temps. Finalement, la couche applicative regroupe les applications industrielles ciblées.

applications contraintes	applications non contraintes		couche applicative
Rocari	EOLSR		couche réseau
Macari (TDMA)	Macari (CSMA/CA)	Serena	couche liaison
Macari (gestion du temps)			
PHY 802.15.4			couche physique

Figure 3.1 – La pile protocolaire Ocari.

La topologie considérée est un ensemble d'îlots. Chaque îlot fonctionne sur un canal radio spécifique et peut supporter jusqu'à une centaine de nœuds réseaux. Au sein de chaque îlot, deux types de nœuds sont considérés : les coordinateurs (équivalents aux routeurs de ZigBee) et les feuilles (équivalentes aux feuilles de ZigBee), comme indiqué sur la figure 3.2. Nous considérons que les capteurs et actionneurs sont connectés aux feuilles principalement. Dans chaque îlot, un coordinateur particulier, le coordinateur du PAN (CPAN, équivalent au coordinateur de ZigBee), est responsable de la gestion du réseau et joue le rôle de puits applicatif¹. Le CPAN construit un arbre logique dont il est la racine, à la manière de la topologie en arbre de ZigBee. Cet arbre est utilisé pour réaliser l'adressage des nœuds, pour simplifier leur synchronisation et pour router le trafic contraint en temps.

3.1.2 Sous-couche MAC d'Ocari

La sous-couche MAC d'Ocari est constituée de deux protocoles : le protocole Macari et le protocole Serena². Dans cette partie, nous allons décrire chacun de ces protocoles, puis des optimisations.

1. Plus précisément, chaque CPAN est connecté *via* un réseau filaire au véritable puits applicatif.

2. Traditionnellement, Serena est introduit comme un protocole de la couche réseau. Dans ce document, nous avons choisi de le présenter comme un protocole de la sous-couche MAC, car il

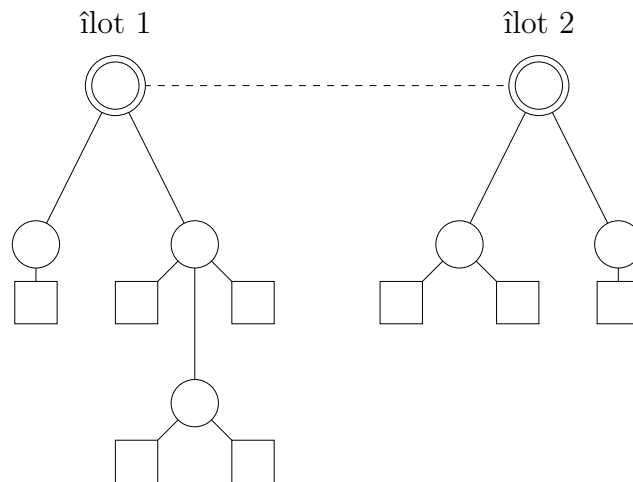


Figure 3.2 – Une topologie Ocarri avec deux îlots. Les coordinateurs sont représentés par des ronds et les feuilles par des carrés. Les deux CPAN sont représentés par des coordinateurs ayant un cercle supplémentaire.

3.1.2.1 Le protocole Macari

Le protocole Macari (signifiant MAC pour Ocarri) [CGM08*] a été développé par le LIMOS et par le LATTIS. Macari découpe le temps en un cycle global de quatre périodes : une période de synchronisation $[T_0; T_1]$, une période d'activités ordonnancées $[T_1; T_2]$, une période d'activités non ordonnancées $[T_2; T_3]$, et une période d'inactivité $[T_3; T_0]$. Ce découpage est présenté sur la figure 3.3.

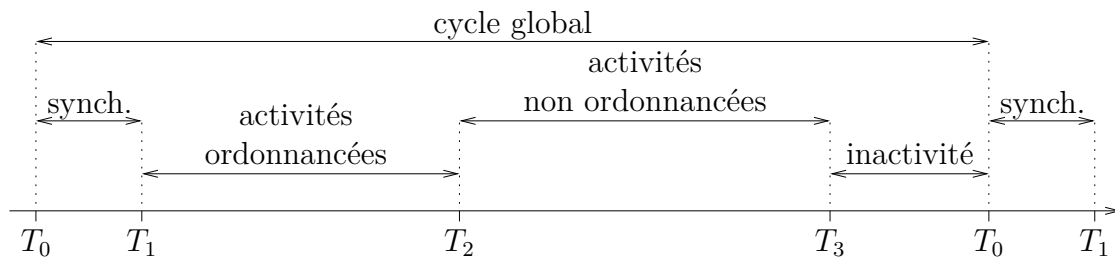


Figure 3.3 – Macari découpe le temps en un cycle (appelé cycle global) de quatre périodes.

La période de synchronisation $[T_0; T_1]$.

La période de synchronisation permet à tous les nœuds du réseau de se synchroniser sur une vision commune du temps, donnée par le CPAN. Pendant cette période, le CPAN envoie une balise qui est répétée à tour de rôle par chaque coordinateur du réseau. L'ordre avec lequel les coordinateurs envoient la balise est appelé l'ordre de synchronisation. La seule contrainte sur cet ordre est qu'il soit topologique : tout coordinateur doit apparaître dans l'ordre avant ses fils.

influe sur les aspects temporels (dévolus à la sous-couche MAC) et pas sur les aspects routage (dévolus à la couche réseau).

La figure 3.4 présente les actions effectuées par les coordinateurs pendant la période de synchronisation, en prenant comme exemple la topologie représentée sur la figure 3.5, sur l'ordre de synchronisation (a, b, c, d, e) . Tout d'abord, a prépare la balise au niveau de la sous-couche MAC et la confie au module radio. Une fois prise en charge par le module radio, la balise est émise par a . La balise est reçue simultanément³ par toutes les feuilles de a (non représentées ici) et par les trois coordinateurs fils de a (qui sont b, c et d). Pour tous ces nœuds, la trame est confiée à la sous-couche MAC, puis analysée. Après le traitement, tous ces nœuds sont synchronisés par l'instant d'arrivée de la trame. Les coordinateurs b, c et d doivent retransmettre la trame pour leurs feuilles et pour leurs coordinateurs fils. En examinant l'ordre de synchronisation contenu dans la balise, b détermine qu'il est le prochain coordinateur à transmettre la balise. Il la confie donc au module radio, qui émet la trame. Les feuilles du coordinateur b la reçoivent (la balise est ignorée par le nœud a et par les nœuds qui la reçoivent plus d'une fois). Parallèlement, c détecte que c'est au tour de b de transmettre, et retarde donc sa transmission (la durée du retard est égale au temps nécessaire pour confier la trame au module radio, pour l'émettre, pour la confier à la sous-couche MAC du côté des récepteurs, et pour la traiter). Après cette attente, c confie la trame au module radio. Après réception de la balise de a , le coordinateur d effectue la même procédure que c , mais avec un délai double, puisque d est le deuxième coordinateur après b dans l'ordre de synchronisation. Lorsque d envoie sa trame, e la reçoit. Le module radio de e confie la trame à la sous-couche MAC, puis e la traite, la confie au module radio, puis l'émet. Une fois que toutes les feuilles de e ont eu le temps de confier la trame à leur sous-couche MAC et de la traiter, la période de synchronisation est terminée : tous les nœuds du réseau connaissent l'instant T_1 .

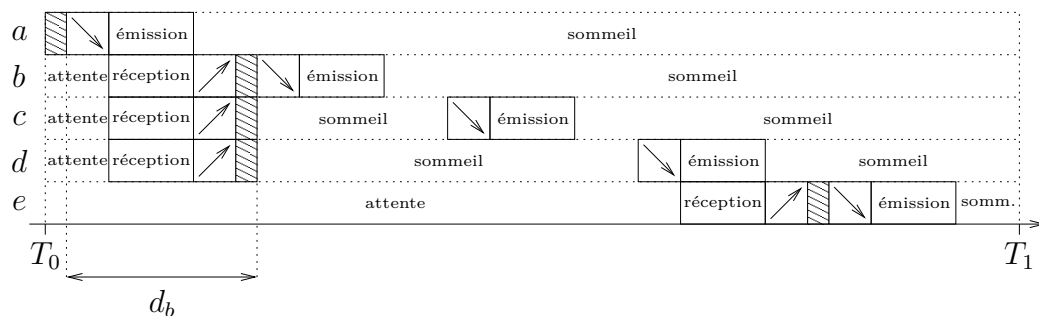


Figure 3.4 – Pendant $[T_0; T_1]$, chaque coordinateur envoie une balise à tour de rôle, en commençant par le CPAN. Les zones hachurées (d'une durée de t_b) représentent le traitement de la balise au niveau de la sous-couche MAC, les zones avec des flèches descendantes représentent l'action de confier une trame au module radio, et les zones avec des flèches montantes représentent l'action de confier une trame à la sous-couche MAC.

La figure 3.4 présente aussi les intervalles de temps pendant lesquels le module radio de chaque nœud peut être mis en sommeil. Pendant $[T_0; T_1]$, chaque coordinateur est en sommeil entre la fin du traitement de la balise reçue de son père et le

3. Le délai de propagation est négligeable pour les distances concernées.

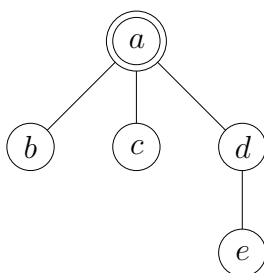


Figure 3.5 – Une topologie logique de cinq coordinateurs, a étant le CPAN.

début de l’envoi de sa balise (si cet intervalle de temps est suffisant pour éteindre le module radio puis le rallumer), et entre la fin de l’envoi de sa balise et l’instant T_1 . Les feuilles sont en sommeil entre la fin du traitement de la balise reçue de leur père et l’instant T_1 . Tous les nœuds sont en attente entre T_0 et le début de la réception de la balise de leur père : pendant cet intervalle de temps, ils écoutent le médium, ce qui consomme de l’énergie.

L’économie d’énergie moyenne réalisée pendant la période de synchronisation est proche de 50%. Pour s’en convaincre, calculons le taux d’activité selon les hypothèses suivantes : la consommation en mode sommeil est négligeable (par rapport à la consommation dans les autres modes), et tous les autres états (comme l’écoute, la transmission et la réception) consomment de manière similaire⁴. Tout d’abord, notons n le nombre de coordinateurs et $m = c.n$ le nombre de feuilles (en supposant que chaque coordinateur possède c feuilles). Notons d_b la durée nécessaire pour qu’un coordinateur confie la balise au module radio, pour transmettre la balise, pour qu’un coordinateur ayant reçu la balise la confie à la sous-couche MAC, et pour qu’il la traite (voir sur la figure 3.4). La durée totale de la période $[T_0; T_1]$ est donc $n.d_b + t_b$, où t_b est le temps de traitement du CPAN avant l’envoi initial de la balise. La durée d’attente du i -ème coordinateur de l’ordre de synchronisation est $(i - 1).d_b + t_b$, avec $i \in [1; n]$. Après cette attente, chaque coordinateur utilise d_b unités de temps pour recevoir, traiter et transmettre sa balise. Ainsi, la somme des durées d’activité des coordinateurs est $\sum_{i=1}^n ((i - 1).d_b + t_b) + \sum_{i=1}^n d_b = n.(d_b.(n + 1) + 2t_b)/2$. Comme la somme des durées de $[T_0; T_1]$ pour les coordinateurs est $n.(n.d_b + t_b)$, on peut en déduire que le taux d’activité des coordinateurs est $n.(d_b.(n + 1) + 2t_b)/(2n.(n.d_b + t_b)) = \frac{1}{2} + \frac{d_b + t_b}{2.d_b.n + 2t_b}$: les coordinateurs sont donc actifs en moyenne un peu plus de la moitié du temps (avec toutefois une grande disparité entre les coordinateurs). Pour les feuilles, on obtient le même résultat.

Nous avons évalué, par simulation, la durée de la période $[T_0; T_1]$ en fonction de la topologie dans [CGJ⁺08*, CLG⁺09*]. Pour cela, nous avons fixé un temps de traitement des balises de 0,08 ms, nous avons considéré que les nœuds étaient déployés aléatoirement, et nous avons produit des topologies en arbre ayant des paramètres différents. Les durées obtenues sont les suivantes.

- Quand la profondeur des arbres est 3 et que le nombre de fils par nœud varie

4. Par exemple, sur le composant CC2420 [CC2420], la consommation en mode sommeil est 426 μA (dans le sommeil le moins profond), la consommation en écoute et en réception est de 19,7 mA, et la consommation en transmission varie entre 8,5 mA et 17,4 mA (selon la puissance de transmission).

entre 3 et 5, la durée de $[T_0; T_1]$ varie entre 0,035 s pour 10 nœuds et 0,19 s pour 24 nœuds [CGJ⁺08*].

- Quand la profondeur des arbres est 4 et que le nombre de fils par nœud est de 3 au maximum (ou que la profondeur des arbres est 6 et que le nombre de fils par nœud est de 2 au maximum), la durée de $[T_0; T_1]$ varie entre 0,14 s pour 20 nœuds et 0,8 s pour 50 nœuds [CLG⁺09*].

L'avantage de cette période de synchronisation est qu'elle garantit que tous les nœuds du réseau peuvent se synchroniser (du moment qu'ils sont à portée d'un coordinateur) sans provoquer de collision de balises. L'inconvénient est que cette méthode ne passe pas à l'échelle (la balise ne peut pas contenir plus d'une cinquantaine d'adresses de coordinateurs car sa taille est limitée) et que la durée de la synchronisation croît avec le nombre de coordinateurs dans le réseau (d'une manière non linéaire, à cause de l'augmentation de la taille des balises). Toutefois, l'impact de ces inconvénients peut être réduit de la manière suivante.

- Le passage à l'échelle peut être amélioré en compressant l'ordre de synchronisation. Par exemple, le mécanisme MBC (pour *Macari beacon compression*, [AACG⁺09*]) profite des propriétés de l'adressage hiérarchique utilisé pour rendre certaines adresses implicites dans l'ordre, et ainsi réduire la taille de la liste.
- La durée totale de la période de synchronisation est faible par rapport à la durée du cycle global, même lorsque le nombre de coordinateurs dans le réseau approche de la cinquantaine. D'après les résultats de nos prototypes (pour un nombre réduit de coordinateurs), nous avons $t_b \approx 2$ ms et $d_b \approx 4$ ms. En extrapolant pour 50 coordinateurs, on peut considérer que la durée de $[T_0; T_1]$ pour 50 nœuds est d'environ 202 ms⁵. Nous décrivons dans la partie 3.1.2.3 une heuristique permettant de réduire davantage cette durée en profitant du parallélisme de l'exécution des nœuds.

La période d'activités ordonnancées $[T_1; T_2]$.

La période d'activités ordonnancées permet à tous les nœuds du réseau de communiquer en limitant les risques d'interférences. Un intervalle de temps est alloué pour chaque coordinateur ainsi que pour ses feuilles : aucun autre coordinateur du réseau ne peut communiquer pendant cet intervalle de temps (à l'exception du père du coordinateur, sur sollicitation du fils). Ces intervalles de temps séquentiels sont représentés par un ordre, dit ordre d'activités. Chacun de ces intervalles de temps est décomposé en trois parties : un temps de communication entre le coordinateur et ses feuilles, un temps de communication entre le coordinateur et son père, et un temps de communication entre le père du coordinateur et le coordinateur lui-même.

La figure 3.6 présente les actions effectuées par les coordinateurs pendant la période d'activités ordonnancées, en prenant comme exemple la topologie représentée sur la figure 3.5, pour l'ordre d'activités (a, b, c, d, e) . Tout d'abord, a communique avec ses feuilles (a étant le CPAN, il n'a pas de père et son intervalle de temps entier

5. Il s'agit ici d'un ordre de grandeur, étant donné que le prototype n'a pas été testé pour un grand nombre de coordinateurs. Il est probable que le temps de traitement d'une balise contenant 50 coordinateurs dépasse la valeur de 2 ms.

peut être réservé pour ses feuilles). À la fin de son intervalle de temps, a cesse de dialoguer avec ses feuilles. Au début de l'intervalle de temps alloué à b , b dialogue avec ses feuilles. Puis, avant la fin de son intervalle de temps, b communique avec son père a , qui était en attente, pour lui envoyer des trames. Ensuite, et sur sollicitation de b , le père a communique avec b jusqu'à la fin de l'intervalle de temps alloué pour b . Pendant ces échanges entre père et fils, aucun autre nœud du réseau ne communique (y compris les feuilles de b). Ce procédé se répète pour c , d et e .

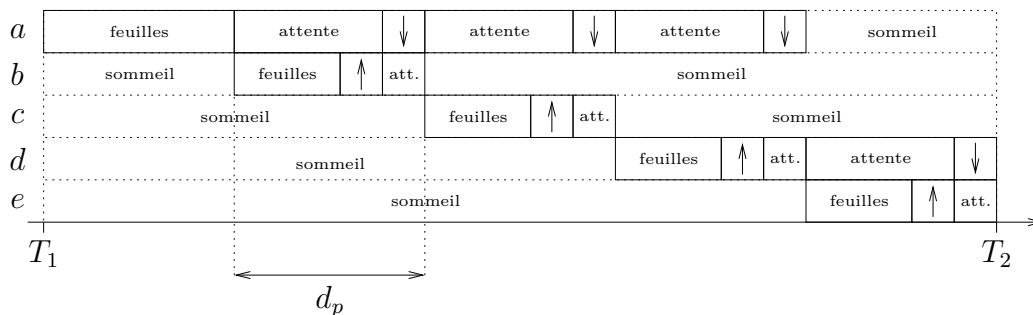


Figure 3.6 – Pendant $[T_1; T_2]$, un intervalle de temps est alloué pour chaque coordinateur ainsi que pour ses feuilles. Les zones avec des flèches vers le haut représentent les communications d'un fils vers son père, et les zones avec des flèches vers le bas représentent les communications d'un père vers l'un de ses fils.

L'accès au médium diffère selon chaque partie de l'intervalle de temps.

- La gestion des communications entre un coordinateur et ses feuilles est proposée principalement par le LATTIS dans [LvdBV08]. Le mécanisme utilisé inclut une sous-période où l'accès est basé sur CSMA/CA slotté et une sous-période avec des GTS. Les spécificités de la proposition sont de permettre plusieurs niveaux de réservation des GTS, et de permettre l'allocation de GTS simultanés en tirant partie de l'effet capture (temporel et de puissance) pour éviter les collisions [vdB07].
- La gestion des communications du fils au père et du père au fils se fait grâce au passage d'un droit de parole, qui est implicitement attribué au fils au début de son intervalle de temps.

La période d'activités ordonnancées permet de fournir un délai borné de bout en bout, sous certaines conditions. Cette propriété se base sur l'hypothèse suivante : même si la durée de $[T_0; T_1]$ et de $[T_1; T_2]$ dépend du nombre de coordinateurs dans le réseau, le temps de cycle a une durée constante ; à mesure que le nombre de coordinateurs augmente (et donc que $[T_0; T_1]$ et $[T_1; T_2]$ augmentent), la durée de la période d'inactivité $[T_3; T_0]$ réduit d'autant. Les constats permettant de borner le délai de bout en bout sont les suivants.

- Lorsque l'ordre d'activités est descendant, c'est-à-dire lorsqu'il s'agit d'un ordre topologique partant du CPAN (par exemple, s'il s'agit de (a, b, c, d, e) comme sur l'exemple choisi pour la figure 3.6), il est possible de transmettre une trame d'une feuille d'un coordinateur quelconque à n'importe quel coordinateur de sa descendance avant T_2 . En revanche, il n'est possible de transmettre une trame d'une feuille d'un coordinateur quelconque qu'au coordinateur père avant T_2 .

- Lorsque l'ordre d'activités est montant, c'est-à-dire lorsqu'il s'agit de l'inverse d'un ordre topologique partant du CPAN (par exemple, (e, d, c, b, a)), il est possible de transmettre une trame d'une feuille d'un coordinateur quelconque à n'importe lequel de ses coordinateurs ancêtres avant T_2 . En revanche, il n'est possible de transmettre une trame d'une feuille d'un coordinateur quelconque qu'à l'un de ses coordinateurs fils (directs) avant T_2 .

En alternant l'ordre d'activités entre montant et descendant à chaque cycle global, il est possible de transmettre une trame d'une feuille f_x d'un coordinateur x à une feuille f_y d'un coordinateur y , en un délai d'au plus quatre cycles, si le réseau n'est pas chargé (c'est-à-dire, si les trames prévues pour chaque intervalle peuvent être envoyées pendant cet intervalle). En effet, nous avons dans le pire des cas :

- au premier cycle, un ordre d'activité descendant : la feuille f_x transmet la trame à son coordinateur x , qui la transmet à son coordinateur père,
- au deuxième cycle, un ordre d'activité montant : le coordinateur père de x transmet la trame au coordinateur z , ancêtre commun de x et y sur l'arbre, quelque soit la distance entre x et z sur l'arbre,
- au troisième cycle, un ordre d'activité descendant : le coordinateur z transmet la trame au coordinateur y , qui est l'un de ses descendants sur l'arbre, quelque soit la distance entre z et y sur l'arbre,
- au quatrième cycle, le coordinateur y transmet la trame à sa feuille f_y .

Cette borne sur le délai de bout en bout n'est applicable que si la durée des communications entre père et fils est suffisamment grande pour transmettre toutes les trames contraintes, et impose que le routage suive l'arbre⁶.

L'économie d'énergie moyenne réalisée pendant la période d'activités ordonnées est très importante. Pour les n coordinateurs, la durée totale d'activité est $\sum_{i=1}^n d_p + \sum_{i=1}^{n-1} d_p$, où d_p correspond à la durée d'un intervalle de temps (voir sur la figure 3.6). Le taux d'activité des coordinateurs, c'est-à-dire la proportion du temps où les coordinateurs ne dorment pas, est donc $\frac{2n \cdot d_p - d_p}{n^2 \cdot d_p} = \frac{2}{n} - \frac{1}{n^2}$. Pour les m feuilles, le taux d'activité est $\frac{m \cdot d_p}{m \cdot n \cdot d_p} = \frac{1}{n}$.

Nous avons évalué, par simulation et par prototypage, les performances de la période $[T_1; T_2]$ pour une étoile [CLG⁺09*]. Le tableau 3.1 présente le temps nécessaire pour que plusieurs nœuds envoient un certain nombre de trames. Les résultats de prototypage montrent que la simulation sous-estime la durée de l'envoi de trames. La différence de temps s'explique principalement par les temps de traitement des trames qui sont mal pris en compte dans la simulation, et par les événements critiques survenant sur le prototype (tels que des interruptions matérielles) qui ralentissent l'exécution des tâches. Le tableau 3.2 présente le taux de réception pour le coordinateur quand un certain nombre de trames est envoyé par plusieurs nœuds. Cette fois, les résultats de simulation sont légèrement inférieurs à ceux obtenus par le prototype. La différence vient probablement des conditions de propagation qui sont très favorables pour le prototype (tous les nœuds se trouvant très proches les uns des autres). Globalement, ces mesures permettent de paramétrer l'environnement de simulation pour qu'il soit plus proche du prototypage (voir la partie 5.2.1).

6. Nous verrons en 3.1.3.1 que cette borne peut être conservée même si le routage est légèrement différent du routage hiérarchique, comme c'est le cas pour Rocari.

Nombre de nœuds	Nombre de trames envoyées (par nœud)	Durée pour la simulation	Durée pour le prototype
1	1	0,005 s	0,013 s
1	3	0,013 s	0,026 s
1	5	0,023 s	0,040 s
3	1	0,017 s	0,025 s
3	3	0,039 s	0,052 s
3	5	0,062 s	0,083 s
5	1	0,023 s	0,033 s
5	3	0,053 s	0,075 s
5	5	0,088 s	0,113 s

Tableau 3.1 – Comparaison, entre la simulation et le prototypage, du temps nécessaire pour que plusieurs nœuds envoient plusieurs trames [CLG⁺09*].

Nombre de nœuds	Nombre de trames envoyées (par nœud)	Taux de réception pour la simulation	Taux de réception pour le prototype
1	1	100%	100%
1	3	100%	100%
1	5	100%	100%
3	1	100%	100%
3	3	89%	100%
3	5	90%	93%
5	1	97%	100%
5	3	83%	90%
5	5	82%	88%

Tableau 3.2 – Comparaison, entre la simulation et le prototypage, du taux de réception pour le coordinateur quand plusieurs nœuds envoient plusieurs trames [CLG⁺09*].

L'avantage principal de cette période d'activités ordonnancées est qu'elle permet de fournir un délai de bout en bout borné (si la période $[T_1; T_2]$ n'est pas surchargée), quelque soit la taille du réseau. L'inconvénient est que la durée de cette période croît avec le nombre de coordinateurs dans le réseau⁷.

La période d'activités non ordonnancées $[T_2; T_3]$.

La période d'activités non ordonnancées permet aux coordinateurs de communiquer entre eux sans se limiter aux liens de l'arbre. Pendant cette période, les feuilles sont inactives. Cette période est découpée en deux sous-périodes : une sous-période colorée (éventuellement vide) et une sous-période non colorée.

La sous-période colorée se base sur la réutilisation spatiale : certains coordinateurs ont une couleur (qui leur est affectée par le protocole Serena, comme expliqué dans 3.1.2.2), plusieurs coordinateurs pouvant avoir la même couleur. Dans la sous-période colorée, le temps est divisé en intervalles de temps, chacun correspondant à une couleur. Seuls les coordinateurs ayant la couleur correspondant à l'intervalle de temps peuvent transmettre des données pendant cet intervalle de temps, les autres coordinateurs étant uniquement autorisés à transmettre un acquittement.

Dans la sous-période non colorée, tous les coordinateurs peuvent parler. Les nouveaux nœuds qui souhaitent rejoindre le réseau envoient leurs requêtes pendant cette sous-période (qu'ils soient coordinateurs ou feuilles)⁸.

L'économie d'énergie durant la période d'activités non ordonnancées est maximale pour les feuilles, puisqu'elles sont inactives. Pour les coordinateurs, l'économie d'énergie est réalisée uniquement pendant la période colorée. Un coordinateur peut envisager de passer en mode sommeil dans tous les intervalles de temps qui correspondent à une couleur qui n'est ni la sienne (dans ce cas, il ne sera pas émetteur), ni celle de l'un de ses voisins (dans ce cas, il ne sera pas récepteur). Toutefois, il faut noter que le passage en mode sommeil prend un temps non négligeable⁹ : si la durée de sommeil envisagée est trop courte, le coordinateur ne peut pas désactiver son module radio pour économiser de l'énergie.

L'accès au médium pendant la période d'activités non ordonnancées se fait en utilisant l'algorithme CSMA/CA slotté¹⁰. Il est possible d'utiliser cet algorithme grâce à la synchronisation des nœuds sur l'instant T_1 . Toutefois, à cause de la dérive des horloges des nœuds, la durée entre T_1 et T_3 ne doit pas être trop grande¹¹.

7. Pour donner un ordre de grandeur, nous considérons dans [ERCGM11*] que 32 ms sont attribués dans $[T_1; T_2]$ par coordinateur. Pour un réseau de 40 coordinateurs, cela revient à réserver 1,28 s pour $[T_1; T_2]$.

8. Pour avoir le droit d'émettre ces requêtes, les nouveaux nœuds ont donc dû recevoir préalablement la balise de synchronisation (pendant $[T_0; T_1]$), et l'avoir traitée pour identifier la sous-période non colorée de $[T_2; T_3]$.

9. Par exemple, le composant CC2420 possède trois modes de sommeil (allant d'un sommeil profond à un sommeil léger). En sommeil profond, le composant met 2,4 ms à se réveiller, alors qu'en sommeil léger, le composant ne met que 30 μ s à se réveiller [JRO10]. Les résultats obtenus sur les composants FreeScale [FreeScale] dans [FvdBV12] indiquent que le composant radio met 310 μ s à se réveiller et que le microcontrôleur met 235 μ s à se réveiller.

10. Dans les optimisations détaillées en 3.1.2.3, nous verrons qu'il est possible d'utiliser un autre algorithme, TDMA/CA, pour l'accès au médium dans la sous-période colorée.

11. Lorsque la durée entre T_1 et T_3 dépasse 4 secondes, la dérive des horloges entre deux nœuds

L'avantage de la période d'activités non ordonnancées est que le routage est plus efficace que dans $[T_1; T_2]$, puisque les communications ne sont plus limitées à l'arbre. Les inconvénients principaux sont les suivants : la sous-période colorée nécessite l'échange de plusieurs messages de contrôle (pour affecter une couleur aux coordinateurs), et la sous-période non colorée met en compétition tous les nœuds pour l'accès au médium.

La période d'inactivité $[T_3; T_0]$.

La période d'inactivité permet à tous les nœuds d'économiser de l'énergie en éteignant leur module radio. La période d'inactivité commence à T_3 , et termine avant l'envoi de la prochaine balise par le CPAN¹². Cette période d'inactivité a été introduite dans [AACG⁺09*] : avant, l'économie d'énergie se faisait exclusivement pendant les autres périodes.

La durée de la période d'inactivité dépend de la durée du cycle global, contenue dans chaque balise. La période d'inactivité pouvant être très longue (si les besoins de réactivité de l'application le permettent), la dérive des nœuds peut avoir un effet important. Chaque nœud doit donc considérer sa dérive maximum avec le CPAN, afin de se réveiller avant que T_0 n'ait commencé. Désignons par d_g la durée du cycle global (entre un T_0 et le T_0 suivant), par δ_r le temps de réveil, par $d_{t_0t_1}$ la durée de l'intervalle $[T_0; T_1]$, et considérons une dérive d'au plus Δ ppm par nœud. Le CPAN attend d_g unités de temps entre chaque balise (selon son horloge, qui est utilisée comme référence dans le réseau). Chaque autre nœud considère que sa dérive est Δ (c'est-à-dire que pour le nœud, le temps s'écoule lentement) et que la dérive du CPAN est $-\Delta$ (c'est-à-dire que pour le CPAN, le temps s'écoule rapidement). Dans ces conditions, le nœud doit initier son réveil au plus tard après $(d_g - d_{t_0t_1} - \delta_r)(1 - 2\Delta)$ unités de temps après le T_1 précédent.

La période d'inactivité induit une accumulation du trafic dans les files d'attente. En conséquent (lorsque le réseau n'est pas surchargé), le début des périodes d'activité des nœuds est plus chargé que la fin des périodes d'activité. Nous avons étudié pendant la thèse de Nassima HADID l'impact de l'inactivité sur la période d'activité [CHGM09*], et nous avons proposé une modification de l'algorithme CSMA/CA pour prendre en compte ce phénomène [HGM09a*, HGM09b*].

3.1.2.2 Le protocole Serena

Le protocole Serena (*Schedule Router Nodes Activity*) [MM08b] a été développé par Inria. Serena est un protocole qui attribue à chaque coordinateur une couleur unique dans un voisinage à trois sauts, comme représenté sur la figure 3.7 (les couleurs sont représentées comme des numéros sous les nœuds). Par exemple, les nœuds a , d et j , éloignés d'au moins trois sauts les uns des autres, peuvent partager la

peut dépasser la durée des intervalles de garde de Macari, qui est de $320 \mu\text{s}$, en considérant une dérive Δ par nœud de 40 ppm. En effet, la dérive maximale entre deux nœuds étant 2Δ , on obtient $4 * 2 * 40 \cdot 10^{-6} = 320 \cdot 10^{-6}$ s, soit $320 \mu\text{s}$.

12. On peut noter que les nœuds n'ont généralement pas une connaissance précise de T_0 . En effet, les nœuds se synchronisent sur T_1 , et lorsque le prochain T_0 arrive, ils sont considérés comme désynchronisés.

couleur 1. Les couleurs affectées par Serena sont ensuite utilisées dans Macari pendant la sous-période colorée de la période d'activités ordonnancées $[T_2; T_3]$, afin de faire de la réutilisation spatiale en déterminant quels coordinateurs peuvent parler simultanément.

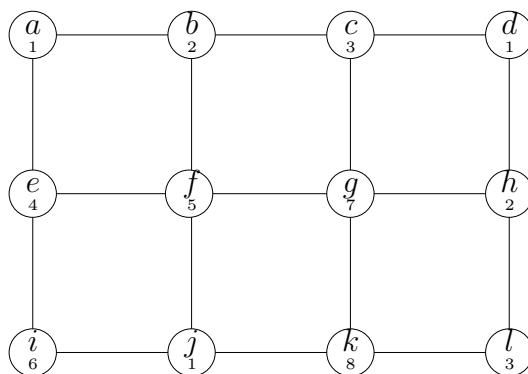


Figure 3.7 – Dans Serena, deux nœuds séparés de trois sauts ou plus (comme a et d) peuvent réutiliser la même couleur.

L'algorithme de coloriage de Serena nécessite que chaque coordinateur connaisse son voisinage à trois sauts, et permet à chaque coordinateur de s'allouer une couleur de manière distribuée. La connaissance du voisinage à trois sauts est obtenue de la manière suivante. Tout d'abord, chaque nœud connaît son voisinage à un saut par l'intermédiaire d'une table mise à jour par Macari pendant la période $[T_0; T_1]$, car chaque coordinateur reçoit une balise des coordinateurs voisins pendant cette période¹³. Chaque coordinateur propage son voisinage à un saut à ses voisins immédiats, qui peuvent ainsi calculer leur voisinage à deux sauts. Puis, chaque coordinateur propage son voisinage à deux sauts à ses voisins immédiats, qui peuvent ainsi calculer leur voisinage à trois sauts.

L'allocation des couleurs est réalisée de la manière suivante. Chaque couleur est associée à un numéro. Un nœud ne peut s'allouer une couleur que s'il est le nœud de plus haute priorité n'ayant pas encore de couleur dans son voisinage à trois sauts : la couleur choisie par ce nœud est la plus petite couleur non utilisée dans le voisinage à trois sauts. La priorité est basée sur le nombre de nœuds dans le voisinage à trois sauts, et sur l'adresse du nœud.

L'avantage de Serena (lorsqu'il est couplé à une méthode d'accès comme Macari) est qu'il permet à tous les coordinateurs ayant la même couleur de communiquer en parallèle, en réduisant les risques de collision (puisque les coordinateurs ayant la même couleur sont censés être distants). Les inconvénients de Serena incluent le temps nécessaire au coloriage, le nombre de couleurs qui peut être important par rapport au nombre de nœuds quand le réseau est constitué d'un petit nombre de coordinateurs (ce qui est le cas dans la version de base de Macari qui supporte jusqu'à 50 coordinateurs)¹⁴, et le fait que les sauts ne reflètent pas précisément la

13. Ce fonctionnement empêche toutefois aux coordinateurs de dormir pendant la période $[T_0; T_1]$.

14. Dans la partie 5.2.6.2 de [Mah10], on peut voir que le nombre de couleurs utilisées pour un réseau aléatoire de 50 nœuds varie entre 32,5 (quand la densité moyenne du réseau est 10) à 48,6 (quand la densité moyenne est 20).

distance entre les nœuds¹⁵.

3.1.2.3 Optimisations

Dans cette partie, nous allons présenter deux optimisations de la sous-couche MAC du projet Ocari. La première optimisation consiste à réduire la durée de $[T_0; T_1]$ en utilisant un ordre de synchronisation particulier. La deuxième optimisation consiste à réduire la durée d'accès au médium dans la sous-période colorée de $[T_2; T_3]$ en se basant sur le fait que Serena a alloué des couleurs différentes par nœuds dans un même voisinage.

Réduction de la durée de synchronisation.

Il est important de chercher à réduire la durée de $[T_0; T_1]$. En effet, cette période n'est pas utilisée pour l'envoi de données, et constitue donc une surcharge pour le protocole Macari.

Anticipation des transmissions. Une première étape dans la réduction de la durée de $[T_0; T_1]$ consiste à remarquer que pour un ordre de synchronisation donné, certains nœuds peuvent anticiper la transmission de leur balise pendant que d'autres nœuds sont en train de traiter la balise reçue [CLG⁺09*]. Cette optimisation fait l'hypothèse que le temps de traitement d'une balise (par la sous-couche MAC) n'est pas négligeable, ce qui est le cas en pratique. En effet :

- Si l'on considère une balise de 100 octets (au niveau physique), le temps de transmission de ces octets sur le médium est de $3125 \mu\text{s}$. Le temps de traitement d'une telle balise dure environ 1 ms, selon l'étude [CLG⁺09*] faite sur un prototype. Ce temps de traitement correspond à l'exécution d'environ 4000 instructions par le processeur cadencé à 4 MHz (ce traitement inclut le parcours de la cinquantaine d'adresses de coordinateurs présents dans la balise).
- Sur le matériel que nous avons utilisé dans le projet Ocari, propre au partenaire Telit, le module radio (qui est un composant CC2420) est connecté au microprocesseur (qui est chargé d'exécuter la sous-couche MAC) par une liaison SPI (*Serial Peripheral Interface*) à 500 kbps. Le temps d'échange d'une trame entre le module radio et le microprocesseur est donc seulement deux fois plus rapide que le débit théorique de 250 kbps de IEEE 802.15.4. Cette organisation matérielle est similaire sur d'autres plateformes (comme les TelosB [Mem13] par exemple).

La figure 3.8 présente cette optimisation, par rapport à la version non optimisée présentée sur la figure 3.4. Le nœud b ne peut pas anticiper la transmission, puisqu'une fois la balise reçue de a , b est occupé à confier cette balise à la sous-couche MAC, à la traiter, et à confier une nouvelle balise au module radio. Pour le nœud c , en revanche, la situation est différente. Plutôt que d'attendre que les feuilles et coordinateurs fils de b traitent la balise, c peut anticiper la transmission en commençant à confier la balise au module radio pendant que b est en train d'émettre la trame.

15. Il est possible que deux nœuds proches (mais pas à portée l'un de l'autre) ne soient pas voisins à deux sauts à cause de l'absence d'un nœud entre eux.

Le nœud d peut lui aussi anticiper la transmission. En revanche, le nœud e ne peut pas anticiper la transmission (pour la même raison que b).

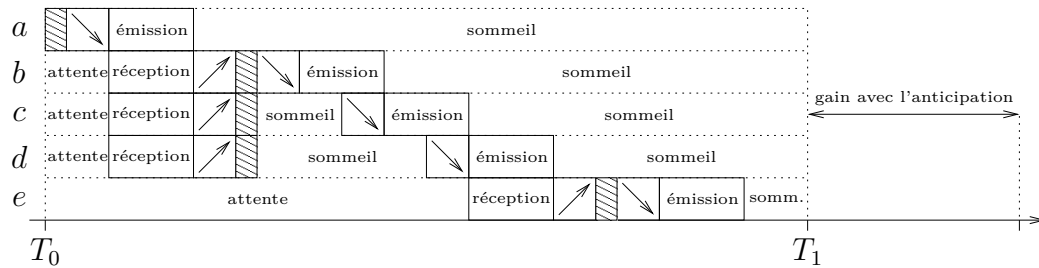


Figure 3.8 – Il est possible de réduire la durée de la période $[T_0; T_1]$ en anticipant les transmissions, pour l'ordre de synchronisation (a, b, c, d, e) et pour la topologie représentée sur la figure 3.5.

Changement de l'ordre de synchronisation. Une deuxième étape dans la réduction de la durée de $[T_0; T_1]$, en complément de la première, consiste à modifier l'ordre de synchronisation afin de maximiser le gain de l'anticipation [DGMD11*]. Pour cela, nous nous basons sur la notion de blocage : un nœud est bloqué s'il ne peut pas anticiper sa transmission. Nous faisons ici l'hypothèse simplificatrice que le temps nécessaire pour confier la balise à la sous-couche MAC, pour traiter la balise, et pour confier la nouvelle trame au module radio, est inférieur au temps d'émission d'une balise. Avec cette hypothèse, on peut remarquer que les nœuds bloqués sont ceux qui suivent immédiatement leur père dans l'ordre (ainsi que le premier nœud)¹⁶.

La figure 3.9 présente cette optimisation, par rapport à la version optimisée présentée sur la figure 3.8. Sur cet exemple, l'ordre (a, d, b, c, e) permet à trois nœuds (b, c et e) d'anticiper leurs transmissions, alors que l'ordre (a, b, c, d, e) de la figure 3.8 ne permettait qu'à deux nœuds (c et d) d'anticiper leurs transmissions. On peut noter que l'ordre (a, d, b, c, e) est l'ordre qui maximise le gain de l'anticipation.

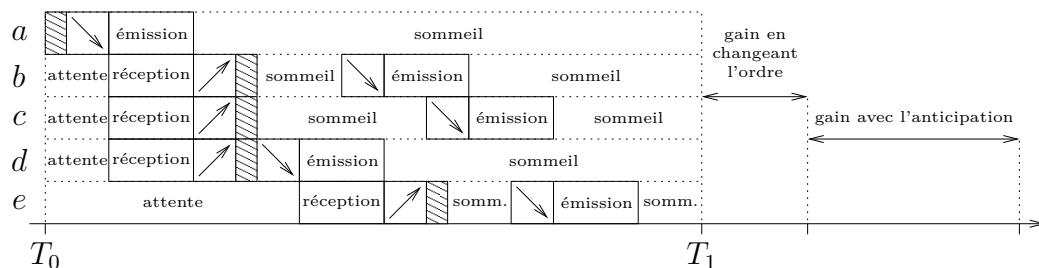


Figure 3.9 – Il est possible de réduire la durée de la période $[T_0; T_1]$ en choisissant l'ordre de synchronisation (a, d, b, c, e) au lieu de (a, b, c, d, e) , pour la topologie représentée sur la figure 3.5.

16. Lorsque l'hypothèse n'est pas vérifiée, un nœud n peut être bloqué si le nombre de nœuds entre son père et lui dans l'ordre est inférieur à un seuil donné. Les algorithmes décrits plus loin dans cette partie pourraient être adaptés à ce cas de figure, au détriment de leur clarté.

Deux algorithmes de calcul de l'ordre optimal. Nous pouvons calculer l'ordre optimal, c'est-à-dire maximisant le gain de l'anticipation, en modélisant le problème au moyen d'un programme linéaire en nombres entiers [DGMD11*]. Ce programme nécessite la connaissance de l'arbre T et de l'ensemble V des nœuds du graphe (le nombre de nœuds est $|V|$), ainsi qu'une grande puissance de calcul¹⁷. Notons père(n) le nœud père de n dans T , et pos(n) la position du nœud n dans l'ordre. permut(n, p) est une matrice booléenne qui indique si le nœud n est en position p dans l'ordre ou non. $\delta(n_1, n_2)$ est égal à la différence de position des nœuds n_1 et n_2 dans l'ordre, plus $|V|$ (pour garantir que $\delta(n_1, n_2) \geq 0$). blocage(n) indique si un nœud n suit directement son père dans l'ordre (ce qui génère un blocage) ou pas. Finalement, les variables λ_i , qui sont des variables réelles (positives) et les variables x_i , qui sont des variables booléennes, sont utilisés pour modéliser les blocages, comme nous verrons par la suite.

La figure 3.10 présente les contraintes du programme linéaire permettant de produire l'ordre optimal. Les contraintes 1 et 2 garantissent que l'ordre est une permutation des nœuds (en utilisant la matrice permut(n, p)). La contrainte 3 fait correspondre à chaque nœud n sa position pos(n) dans l'ordre. La contrainte 4 garantit que l'ordre est un ordre topologique : un fils ne peut pas transmettre la balise avant son père. La contrainte 5 permet de calculer la valeur de $\delta(n_1, n_2)$ pour tous les couples de nœuds. Les contraintes de 6 à 12 sont utilisées pour modéliser les blocages, qui se représentent avec une fonction non convexe (ce qui empêche une modélisation directe). En effet, un blocage apparaît pour chaque nœud n qui est un successeur immédiat de son père dans l'ordre, c'est-à-dire tel que pos(père(n)) - pos(n) = 1. En d'autres termes, blocage(n) est une fonction constante par morceaux d'une variable entière positive $\delta(n)$, avec $\delta(n) = \text{pos}(\text{père}(n)) - \text{pos}(n) + m$ et m choisi suffisamment grand pour que $\delta(n) \geq 0$ pour tout n . $\delta(n)$ est représenté sur la figure 3.11. Selon la formule 4.18 de [Bil07], on peut modéliser cette fonction au moyen de contraintes linéaires de la manière suivante : blocage(n)(a_3) = 1 et $\forall i \in \{1, 2, 4, 5\}$, blocage(n)(a_i) = 0, avec $a_1 = 0$, $a_2 = m$, $a_3 = m + 1$, $a_4 = m + 2$ et $a_5 = 2m$. Les contraintes 6 à 12 découlent de la modélisation de fonctions en escalier décrite dans [Bil07].

Nous pouvons aussi calculer l'ordre optimal avec un algorithme de type *branch and bound* [DGMD11*], beaucoup plus rapide en pratique. Cet algorithme crée un arbre d'exploration \mathcal{T} , à partir d'un arbre T , de la manière suivante. Chaque nœud de \mathcal{T} est un ordre d'une partie des nœuds de T . La racine de \mathcal{T} est l'ordre (vide) sur l'ensemble vide. Si un nœud de \mathcal{T} est un ordre o sur k nœuds, ce nœud a $|V| - k$ fils : le i -ème fils correspond à l'ordre $o \cdot n_i$, où \cdot représente la concaténation et n_i est le i -ème nœud de V qui n'est pas présent dans o . L'arbre \mathcal{T} possède donc $|V|!$ feuilles, qui correspondent à toutes les permutations possibles sur n nœuds. En énumérant ces feuilles et en calculant le nombre de blocages pour chacune, nous pouvons obtenir l'ordre optimal. Pour éviter de construire l'arbre \mathcal{T} complètement, nous stoppons la construction de \mathcal{T} en chaque nœud qui ne correspond pas à un ordre topologique sur T (puisqu'en ajoutant des nœuds après un ordre non topologique,

17. La puissance de calcul nécessaire interdit l'utilisation des deux algorithmes exacts sur un nœud capteur. Ces algorithmes sont présentés pour évaluer la qualité des solutions obtenues par l'heuristique décrite plus loin.

Minimiser $\sum_{n \in V} \text{blocage}(n)$ sous les conditions suivantes :

- (1) $\forall n \in V, \sum_{p \in [1; |V|]} \text{permut}(n, p) = 1$
- (2) $\forall p \in [1; |V|], \sum_{n \in V} \text{permut}(n, p) = 1$
- (3) $\forall n \in V, \text{pos}(n) = \sum_{p \in [1; |V|]} (p \cdot \text{permut}(n, p))$
- (4) $\forall n \in V, \text{pos}(n) > \text{pos}(\text{père}(n))$
- (5) $\forall n_1, n_2 \in V^2, \delta(n_1, n_2) = \text{pos}(n_1) - \text{pos}(n_2) + |V|$
- (6) $\forall n \in V, \text{conflict}(n) = \lambda_3(n)$
- (7) $\forall n \in V, \delta(n, \text{père}(n)) = |V|\lambda_2(n) + (|V| + 1)\lambda_3(n) + (|V| + 2)\lambda_4(n) + 2|V|\lambda_5(n)$
- (8) $\forall n \in V, \lambda_1(n) \leq x_1(n)$
- (9) $\forall n \in V, \forall i \in \{2, 3, 4\}, \lambda_i(n) \leq x_{i-1}(n) + x_i(n)$
- (10) $\forall n \in V, \lambda_5(n) \leq x_4(n)$
- (11) $\forall n \in V, \sum_{i \in \{1, 2, 3, 4\}} x_i(n) = 1$
- (12) $\forall n \in V, \sum_{i \in \{1, 2, 3, 4, 5\}} \lambda_i(n) = 1$

Figure 3.10 – Programme linéaire permettant de calculer l'ordre qui maximise le gain de l'anticipation, permettant ainsi de réduire la durée de $[T_0; T_1]$.

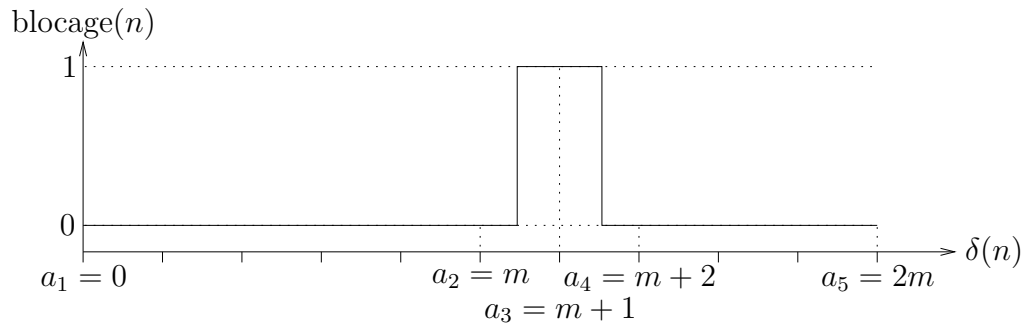


Figure 3.11 – La fonction $\text{blocage}(n)$ est une fonction constante par morceaux de variable $\delta(n)$.

on ne peut pas obtenir un ordre topologique). Nous stoppons aussi la construction de l'arbre en chaque nœud qui correspond à un ordre ayant davantage (ou autant) de blocages que la meilleure solution trouvée (puisqu'en ajoutant des nœuds à un ordre, on ne peut qu'augmenter son nombre de blocages). La solution choisie pour initialiser l'algorithme est un parcours en largeur de l'arbre, qui donne généralement une bonne solution.

Heuristique de calcul d'un ordre approché. Pour calculer rapidement (et avec peu de ressources) un ordre approché ayant peu de blocages, nous avons proposé dans [DGMD11*] une heuristique applicable pour un nœud capteur. L'heuristique part d'un ordre correspondant à un parcours en largeur de l'arbre $B(T)$, tente de réduire les blocages en permutant les nœuds d'une même profondeur, ce qui conduit à un ordre $P(T)$, puis finalement tente de déplacer les nœuds à des profondeurs différentes, ce qui conduit à un ordre $D(T)$. Comme nous allons le voir, le temps de calcul de cette heuristique est linéaire en fonction du nombre de nœuds.

Propriété 1. *Soit un arbre T , et $B(T)$ un parcours en largeur de T . L'ordre $B(T)$ ne contient au plus qu'un blocage par profondeur.*

Démonstration. Comme $B(T)$ est un parcours en largeur, les nœuds dans $B(T)$ sont triés par profondeur croissante. Or, nous avons vu qu'un blocage n'apparaît que pour un nœud qui suit immédiatement son père dans l'ordre. Les blocages dans $B(T)$ ne peuvent donc survenir que lors du passage d'une profondeur à une autre, ce qui arrive au plus h fois, h étant la hauteur de l'arbre T . \square

Pour construire l'ordre $P(T)$, nous partons de $B(T)$. Notons $\text{premier}(d)$ le premier nœud d'un ordre étant en profondeur d de l'arbre, $\text{deuxième}(d)$ le deuxième nœud d'un ordre étant en profondeur d de l'arbre, et $\text{dernier}(d)$ le dernier nœud d'un ordre étant en profondeur d de l'arbre. Pour chaque nœud bloqué à la profondeur $d + 1$ dans $B(T)$, on tente d'appliquer l'une des trois règles de permutation suivantes :

- P_1 : s'il y a au moins trois nœuds en profondeur d , on inverse $\text{deuxième}(d)$ et $\text{dernier}(d)$ dans l'ordre.
- P_2 : s'il y a exactement deux nœuds en profondeur d , et qu'ils ont le même père, on inverse $\text{premier}(d)$ et $\text{dernier}(d)$ dans l'ordre.
- P_3 : s'il y a exactement deux nœuds en profondeur d , et qu'ils ont des pères différents, on inverse $\text{premier}(d)$ et $\text{dernier}(d)$ dans l'ordre, et on applique P_1 , P_2 ou P_3 en profondeur $d - 1$ (pour faire disparaître le blocage que l'on vient d'introduire). On peut remarquer que la règle P_3 peut donc s'appliquer récursivement.

Ces trois règles conservent le fait que l'ordre est un parcours en largeur, car ces règles ne changent pas la profondeur des nœuds.

La figure 3.12 présente l'application de ces trois règles sur trois topologies différentes, en partant du parcours en largeur $B(T) = (a, b, c, d, e, f)$. Pour cet ordre et dans la topologie (1), le nœud e (en profondeur 2) est bloqué, et il existe trois nœuds en profondeur 1 : la règle P_1 peut être appliquée, et l'ordre résultant est obtenu en inversant $\text{deuxième}(1) = c$ et $\text{dernier}(1) = d$, soit (a, b, d, c, e, f) . Le blocage a été

résolu puisqu'il ne reste plus que le nœud b qui est bloqué. Pour l'ordre $B(T)$ et dans la topologie (2), le nœud d (en profondeur 2) est bloqué, et il existe deux nœuds en profondeur 1 de même père : la règle P_2 peut être appliquée, et l'ordre résultant est obtenu en inversant $\text{premier}(1) = b$ et $\text{dernier}(1) = c$, soit (a, c, b, d, e, f) . Le blocage a été résolu puisqu'il ne reste plus que le nœud c qui est bloqué. Pour l'ordre $B(T)$ et dans la topologie (3), le nœud f (en profondeur 3) est bloqué, et il existe deux nœuds en profondeur 2 de pères différents : la règle P_3 peut être appliquée, et l'ordre résultant est obtenu en inversant $\text{premier}(2) = d$ et $\text{dernier}(2) = e$ (ce qui rend le nœud e bloqué), puis $\text{premier}(1) = b$ et $\text{dernier}(1) = c$, soit (a, c, b, e, d, f) .

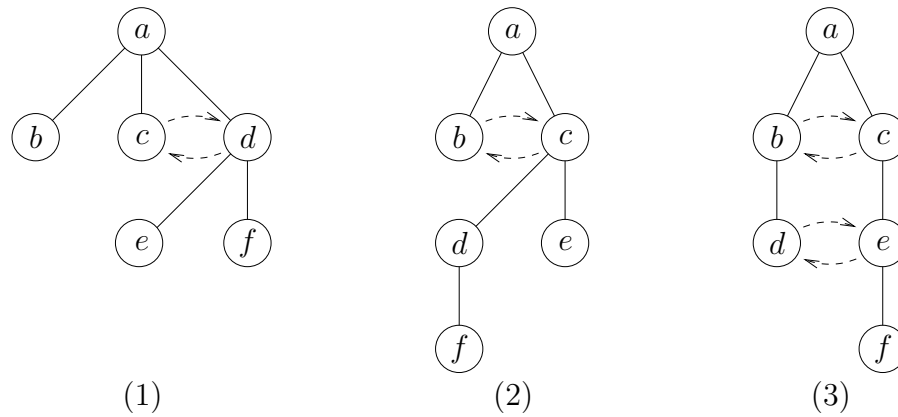


Figure 3.12 – Trois topologies sur lesquelles les trois règles de permutation de l'heuristique s'appliquent.

Propriété 2. *L'application de la règle P_1 réduit de un le nombre de blocages d'un parcours en largeur.*

Démonstration. Soit T un arbre et $B(T)$ un parcours en largeur de T . Supposons que la règle P_1 soit applicable à la profondeur d , c'est-à-dire que le nœud $\text{premier}(d+1)$ est bloqué. L'ordre avant l'application de P_1 peut s'écrire $(\dots, \text{dernier}(d-1), \text{premier}(d), \text{deuxième}(d), \dots, \text{dernier}(d), \text{premier}(d+1), \dots)$. Après application de la règle P_1 , qui échange la position de certains nœuds de profondeur d , le nouvel ordre peut s'écrire $(\dots, \text{dernier}(d-1), \text{premier}(d), \text{dernier}(d), \dots, \text{deuxième}(d), \text{premier}(d+1), \dots)$. Les blocages dans ce nouvel ordre ne peuvent apparaître qu'à des changements de profondeur. Hors, le nœud $\text{premier}(d+1)$ n'est plus bloqué, car il ne suit pas directement son père (qui est $\text{dernier}(d)$). De plus, aucun nouveau blocage n'a été introduit. L'application de la règle P_1 a donc réduit de un le nombre de blocages de l'ordre. \square

Propriété 3. *L'application de la règle P_2 réduit de un le nombre de blocages d'un parcours en largeur.*

Démonstration. Soit T un arbre et $B(T)$ un parcours en largeur de T . Supposons que la règle P_2 soit applicable à la profondeur d , c'est-à-dire que le nœud $\text{premier}(d+1)$ est bloqué. L'ordre avant application de P_2 peut s'écrire $(\dots, \text{dernier}(d-1), \text{premier}(d), \text{dernier}(d), \text{premier}(d+1), \dots)$. Après application de la règle P_2 , le nouvel ordre peut s'écrire $(\dots, \text{dernier}(d-1), \text{dernier}(d), \text{premier}(d), \text{premier}(d+1), \dots)$. Dans ce nouvel ordre, on a :

- le nœud $\text{dernier}(d)$ peut être bloqué par le nœud $\text{dernier}(d-1)$, si $\text{dernier}(d)$ et $\text{premier}(d)$ ont le même père, mais dans ce cas le blocage existait déjà entre $\text{premier}(d)$ et $\text{dernier}(d-1)$,
- le nœud $\text{premier}(d+1)$ n'est plus bloqué, car il ne suit pas directement son père (qui est $\text{dernier}(d)$).

Dans le nouvel ordre, le nœud $\text{premier}(d+1)$ n'est plus bloqué et aucun nouveau blocage n'a été introduit. L'application de la règle P_2 a donc réduit de un le nombre de blocages dans l'ordre. \square

Propriété 4. *L'application de la règle P_3 réduit de un le nombre de blocages d'un parcours en largeur.*

Démonstration. Soit T un arbre et $B(T)$ un parcours en largeur de T . Supposons que la règle P_3 soit applicable à la profondeur d , c'est-à-dire que le nœud $\text{premier}(d+1)$ est bloqué. L'ordre avant application de P_3 peut s'écrire $(\dots, \text{premier}(d), \text{dernier}(d), \text{premier}(d+1), \dots)$. Après application de la règle P_3 (et avant l'application de la règle P_1 , P_2 ou P_3 à la profondeur $d-1$), le nouvel ordre peut s'écrire $(\dots, \text{dernier}(d), \text{premier}(d), \text{premier}(d+1), \dots)$. Dans ce nouvel ordre, le nœud $\text{premier}(d+1)$ n'est plus bloqué, car il ne suit pas directement son père (qui est $\text{dernier}(d)$). Par contre, un blocage a pu être introduit en profondeur $d-1$.

Supposons qu'un blocage a été introduit en profondeur $d-1$, c'est-à-dire que $\text{premier}(d)$ n'était pas bloqué, mais $\text{dernier}(d)$ l'est. Il ne peut pas y avoir un seul nœud en profondeur $d-1$, sinon ce nœud unique serait à la fois le père de $\text{premier}(d)$ et de $\text{dernier}(d)$, et $\text{premier}(d)$ aurait été bloqué. Il existe donc au moins deux nœuds en profondeur $d-1$, et on peut appliquer l'une des règles P_1 , P_2 ou P_3 . Selon les propriétés vues précédemment, ce blocage introduit peut être supprimé (quitte à introduire un blocage à une profondeur inférieure qui pourra être à son tour supprimé).

Dans le nouvel ordre, le nœud $\text{premier}(d+1)$ n'est plus bloqué et aucun nouveau blocage n'a été introduit (il peut rester des blocages, mais ils n'ont pas été introduits par P_3). L'application de la règle P_3 a donc réduit de un le nombre de blocages dans l'ordre. \square

La construction de l'ordre $D(T)$ se fait à partir de l'ordre $P(T)$, en essayant de supprimer les blocages qui n'ont pas été résolus par l'application des règles P_1 , P_2 et P_3 . Jusqu'à présent, les blocages étaient résolus en permutant des nœuds de même profondeur (voir sur la figure 3.12). À présent, nous allons tenter de résoudre les blocages en permutant des nœuds de profondeur différentes. Pour cela, on peut remarquer que les blocages non résolus par les règles P_1 , P_2 et P_3 apparaissent quand le nœud $\text{premier}(d+1)$ est bloqué (par son père $\text{dernier}(d)$), et qu'il n'existe qu'un seul nœud en profondeur d (c'est d'ailleurs le cas du deuxième nœud de tout ordre, toujours bloqué par son père qui est forcément le premier nœud de l'ordre). Dans ce cas, nous déplaçons un nœud disponible pour $\text{dernier}(d)$ pour le placer entre $\text{dernier}(d)$ et $\text{premier}(d+1)$.

Un nœud n_x est dit *disponible* pour un nœud $\text{dernier}(d)$ s'il s'agit d'un nœud sans enfant, et qu'il peut être déplacé plus loin dans l'ordre afin de servir de nœud intermédiaire. Plus précisément, nous considérons qu'un nœud disponible doit satisfaire les conditions suivantes :

- n_x est avant le nœud dernier(d) dans l'ordre,
- n_x n'a pas de fils dans l'arbre T ,
- notons n_a le nœud précédant n_x dans l'ordre, et n_b le nœud suivant n_x dans l'ordre, alors soit n_a est le père de n_x et de n_b ¹⁸, soit n_a n'est ni le père de n_x ni de n_b ¹⁹.

L'ordre $D(T)$ est obtenu en partant de $P(T)$, et en tentant d'appliquer cette règle de déplacement par ordre croissant de profondeurs.

La figure 3.13 présente l'application de la règle de déplacement sur une topologie, en partant de l'ordre $P(T) = (a, b, c, d, e, f)$ qui possède deux blocages : l'un en b (ce qui est inévitable) et l'autre en f . Pour le nœud b (en profondeur 2), la règle de déplacement s'applique car il n'y a qu'un nœud en profondeur 1. Toutefois, il n'existe aucun nœud disponible pour supprimer le blocage de b . Pour le nœud f (en profondeur 3), la règle de déplacement s'applique aussi car il n'y a qu'un nœud (le nœud e) en profondeur 2. On cherche donc l'ensemble des nœuds disponibles pour f , et on trouve l'ensemble $\{c, d\}$. En effet, a n'est pas disponible car il n'est pas précédé d'un nœud, b n'est pas disponible car il a un fils, c est disponible car $n_a = b$ n'est ni le père de $n_x = c$ ni de $n_b = d$, d est disponible car $n_a = c$ n'est ni le père de $n_x = d$ ni de $n_b = e$, et e est pas disponible car il a un fils. On peut donc déplacer c (ou d) pour qu'il soit entre e et f . L'ordre résultant est (a, b, d, e, c, f) , et cet ordre ne possède plus qu'un blocage en b .

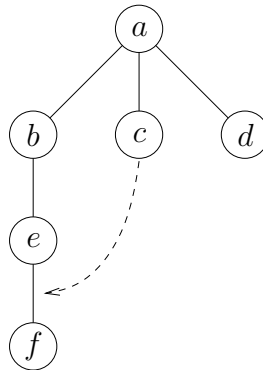


Figure 3.13 – Une topologie sur laquelle la règle de déplacement peut être appliquée pour résoudre le blocage du nœud f .

Propriété 5. *Chaque application de la règle de déplacement réduit de un le nombre de blocages de l'ordre $P(T)$ (éventuellement déplacé).*

Démonstration. Soit T un arbre et $P(T)$ un ordre topologique. Supposons que la règle soit applicable à la profondeur d , c'est-à-dire que le nœud premier($d+1$) est bloqué. L'ordre avant l'application de la règle de déplacement peut s'écrire de la manière suivante : $(\dots, n_a, n_x, n_b, \dots, \text{dernier}(d), \text{premier}(d+1), \dots)$. Après application de la règle de déplacement, le nouvel ordre peut s'écrire $(\dots, n_a, n_b, \dots, \text{dernier}(d), n_x, \text{premier}(d+1), \dots)$. On peut remarquer que :

18. Dans ce cas, n_x était bloqué par n_a . Déplacer n_x conduit au blocage de n_b , ce qui n'augmente pas le nombre de blocages dans l'ordre.

19. Dans ce cas, n_x n'était pas bloqué par n_a . Déplacer n_x ne conduit pas au blocage de n_b , ce qui n'augmente pas le nombre de blocages dans l'ordre.

- Le nouvel ordre reste un ordre topologique (car seul n_x a été déplacé plus loin dans l'ordre, et n_x n'a pas de fils).
- Aucun blocage n'a été introduit. En effet, si n_a était le père de n_x et de n_b , n_x était bloqué dans l'ordre initial : dans le nouvel ordre, c'est à présent n_b qui est bloqué. Si n_a n'était le père ni de n_x ni de n_b , n_x n'était pas bloqué dans l'ordre initial, et n_b n'est pas bloqué dans l'ordre final.
- Le nœud premier($d + 1$) ne suit plus directement son père dernier(d).

Dans le nouvel ordre, le nœud premier($d + 1$) n'est plus bloqué et aucun nouveau blocage n'a été introduit. L'application de la règle de déplacement a donc réduit de un le nombre de blocages dans l'ordre. \square

Propriété 6. *L'heuristique du calcul de $D(T)$ à partir de l'arbre T s'exécute en temps linéaire en fonction du nombre de nœuds de T .*

Démonstration. Posons n le nombre de nœuds de l'arbre T , m le nombre d'arêtes de T (avec $m = n - 1$ car T est un arbre), et h la hauteur de T . Le parcours en largeur $B(T)$ peut être calculé en $\mathcal{O}(n + m) = \mathcal{O}(n)$. L'application d'une permutation P_1 ou P_2 requiert un temps constant. Ainsi, l'application (éventuelle) des permutations P_1 et P_2 à toutes les profondeurs de T requiert $\mathcal{O}(h)$ opérations dans le pire des cas. L'application (éventuelle) de la permutation P_3 à une profondeur donnée implique des opérations concernant les profondeurs inférieures. Toutefois, on peut remarquer que lorsque la règle P_3 est appliquée à deux profondeurs différentes (ou plus), les profondeurs impliquées sont exclusives. Par exemple, si l'on note q_1 et q_2 deux profondeurs où P_3 s'applique, avec $q_1 < q_2$, les profondeurs impliqués par les permutations de la règle P_3 appliquée à q_2 sont toutes supérieures strictement à q_1 . Ainsi, l'application de P_3 à l'ensemble des profondeurs possibles de l'ordre ne nécessite que $\mathcal{O}(h)$ opérations. La complexité pour calculer $P(T)$ est $\mathcal{O}(n + h) = \mathcal{O}(n)$.

L'ensemble \mathcal{D} de tous les nœuds disponibles par rapport au dernier nœud de $P(T)$ peut être calculé en $\mathcal{O}(n)$. Quand l'algorithme cherche un nœud disponible pour le nœud dernier(d), seulement le premier nœud de \mathcal{D} , noté n_x , a besoin d'être considéré²⁰. Si n_x n'est pas disponible pour dernier(d), cela signifie qu'il n'y aura pas d'autres nœuds de \mathcal{D} qui seront disponibles pour dernier(d). Si n_x est disponible pour dernier(d), n_x est déplacé et \mathcal{D} doit être mis à jour : puisque n_x n'a pas d'enfants dans T (à cause de la deuxième contrainte sur les nœuds disponibles), les changements dans \mathcal{D} sont limités aux voisins immédiats de n_x dans l'ordre, à savoir n_a et n_b . En résumé, une fois que l'ensemble \mathcal{D} est calculé (en temps linéaire) pour le dernier nœud de l'ordre, il contient tous les nœuds qui sont potentiellement disponibles, et ne fait que réduire (par au plus n opérations en temps constant). Un nœud qui n'était pas disponible pour un nœud donné ne pourra pas l'être pour un nœud plus éloigné dans l'ordre (c'est-à-dire, plus profond dans l'arbre). En conséquence, calculer $D(T)$ requiert $\mathcal{O}(n)$. L'heuristique complète nécessite donc un temps linéaire pour être appliquée. \square

Résultats. Nous avons évalué, par simulation, le nombre de blocages pour des topologies aléatoires, afin de connaître la durée de $[T_0; T_1]$ avec notre heuristique de

20. Il est important de remarquer ici que l'on ne construit pas un ensemble de nœuds disponibles par nœud, mais un seul ensemble pour tous les nœuds.

réduction des blocages [DGMD11*]. Rappelons que l'arbre est fixé, et que la seule possibilité de l'heuristique est de modifier l'ordre de parcours.

Dans un premier temps, nous avons étudié le nombre de blocages sur des topologies pour lesquelles les nœuds sont déployés aléatoirement (de manière uniforme) dans un plan. Pour chaque topologie, 50 à 2000 nœuds sont déployés. Puis, un arbre est créé dans lequel chaque nœud a au plus 5 fils. Dans ce cas, notre heuristique est optimale dans 99,2% des cas. Ce pourcentage élevé vient du fait que l'arbre produit sur une topologie aléatoire a peu de chances d'avoir un seul nœud à une profondeur donnée, ce qui permet à l'heuristique d'appliquer les règles P_1 , P_2 et P_3 afin d'obtenir un ordre optimal.

Dans un deuxième temps, nous avons étudié le nombre de blocages sur des topologies pour lesquelles les nœuds sont déployés autour de deux zones denses, appelées *hot-spots*, et entre ces zones denses de manière linéaire (afin de se placer dans un cas difficile à traiter pour notre heuristique)²¹. Ces zones denses correspondent à des zones d'intérêt, généralement distantes. Par exemple, dans le cadre de surveillance de barrages hydrauliques le long de longues rivières, il est probable qu'un réseau de capteurs dense soit déployé sur chaque barrage, et qu'un réseau de capteurs linéaire soit déployé entre les barrages. L'arbre construit a de grandes chances d'avoir un seul nœud à plusieurs profondeurs, notamment sur la partie linéaire reliant les deux zones denses. Les performances de notre heuristique restent très bonnes dans ce cas, bien qu'un simple parcours en largeur conduise à de mauvaises performances. Pour les petites topologies (de 20 à 60 nœuds), sur lesquelles il est possible de calculer un ordre optimal, nous avons obtenu une différence très faible entre le nombre de blocages de notre heuristique et le nombre optimal de blocages : en moyenne, la différence est inférieure à 1. Pour les grandes topologies (jusqu'à 8000 nœuds), le nombre de blocages obtenu par notre heuristique reste très faible (en moyenne, il est inférieur à 3), même s'il n'a pas pu être comparé à l'optimal (qui est supérieur ou égal à 1).

Réduction de la durée d'accès au médium pour les activités non ordonnées.

Pour réduire la durée de la sous-période colorée de $[T_2; T_3]$ en profitant du coloriage réalisé par Serena, nous avons proposé dans [MMCG10*] un protocole MAC pour cette sous-période. Le protocole est nommé TDMA/CA (*Time Division Multiple Access with Collision Avoidance*), et réalise un compromis entre TDMA (permis grâce au coloriage) et CSMA/CA. Le protocole est basé sur les remarques suivantes.

- Si le coloriage est correct (par rapport à l'état actuel de la topologie), le fait qu'un seul nœud puisse émettre dans un voisinage à deux sauts permet d'éliminer l'essentiel des risques de collisions²². Dans ce cas, il n'est pas nécessaire d'introduire des retards avant d'envoyer les trames. Cette approche correspond

21. Une approche plus générale aurait été d'utiliser un générateur de topologies plus réalistes, comme celui décrit dans [RGB⁺11].

22. Même avec un coloriage correct, des collisions sont possibles lorsque deux nœuds a et c se trouveraient à deux sauts s'il existait un nœud b entre eux (l'absence de ce nœud intermédiaire b fait que a et c ne se voient pas, et ignorent donc qu'ils sont à deux sauts).

à la partie TDMA du protocole TDMA/CA.

- Si le coloriage est incorrect (suite à la mobilité de certains nœuds, à des changements de l’environnement conduisant à des changements de conditions de propagation, ou à des trames perdues lors du coloriage), des collisions sont possibles. Dans ce cas, on introduit des retards aléatoires avant l’envoi des trames (et TDMA/CA peut informer l’algorithme de coloriage que le coloriage est incorrect). Cette approche correspond à la partie CSMA/CA du protocole TDMA/CA.

L’algorithme 1 décrit la manière dont les trames sont transmises avec l’algorithme TDMA/CA. L’algorithme tente initialement d’envoyer chaque trame t avec un délai de 0, c’est-à-dire sans retard. Si la trame n’est pas acquittée (à cause d’une perte due au médium, ou à cause d’un conflit de couleurs), l’algorithme introduit un retard aléatoire dans l’intervalle $[0; \text{minDélai}]$. À chacune des tentatives suivantes, la taille de la fenêtre de tirage du retard aléatoire double (jusqu’à atteindre un maximum).

Algorithme 1 : Algorithme d’accès au médium selon TDMA/CA.

Entrées : t est une trame à envoyer, maxTentatives , minDélai et maxDélai sont des entiers

$\text{tentatives} \leftarrow 0$

$\text{délai} \leftarrow 0$

tantque t n’est pas acquittée **et** $\text{tentatives} \leq \text{maxTentatives}$ **faire**

 attendre une durée aléatoire dans $[0; \text{délai}]$

 envoyer la trame t

 attendre un acquittement immédiat

si t n’est pas acquittée **alors**

si $\text{tentatives} = 0$ **alors**

$\text{délai} \leftarrow \text{minDélai}$

sinon

$\text{délai} \leftarrow \min\{2.\text{délai}, \text{maxDélai}\}$

finsi

$\text{tentatives} \leftarrow \text{tentatives} + 1$

finsi

fintantque

Le protocole TDMA/CA a les avantages suivants. Lorsqu’il n’y a pas de conflit de couleur, TDMA/CA n’introduit pas de délai supplémentaire. Lorsqu’il y a des conflits de couleur, TDMA/CA peut continuer à transmettre les trames (au prix d’une baisse de performances à cause des délais introduits). De plus, TDMA/CA peut détecter les conflits de couleur et en informer le protocole de coloriage (c’est-à-dire Serena, dans notre exemple). Pour cela, il suffit qu’un nœud récepteur informe Serena lorsqu’il reçoit des trames pour lesquelles il n’est pas la destination, lorsqu’il reçoit des acquittements qu’il n’attend pas, ou lorsque le nombre de tentatives pour envoyer les trames est trop élevé.

Une étude des performances du protocole TDMA/CA, en combinaison avec le coloriage Serena, est présentée dans [MCM⁺10].

3.1.2.4 Bilan de la sous-couche MAC d'Ocari

Les travaux autour de la couche MAC d'Ocari sont nombreux, et ont constitué une thématique importante pour les travaux de recherche de notre équipe pendant plusieurs années. Ces travaux ont aussi conduit à plusieurs collaborations (notamment avec les partenaires de l'IRIT et d'Inria). Une grande partie des travaux de thèse de Nassima HADID a aussi porté sur des optimisations de la sous-couche MAC qui pourraient être appliquées sur Macari [HGM09a*, HGM09b*, GH09*, CHGM09*]. J'ai aussi encadré deux étudiants de master recherche sur cette thématique : Hamid ATTALAH et Hani MAHMOUDI. Ils ont tous les deux travaillé sur l'optimisation de la période $[T_0; T_1]$: Hamid s'est concentré sur la manière dont la dérive des nœuds pouvait être prise en compte de manière analytique, et Hani a conçu un protocole de synchronisation distribué.

Les thématiques autour de la sous-couche MAC dans les réseaux de capteurs sans fil industriels restent au cœur des activités de notre équipe. Les deux perspectives les plus prioritaires de cette thématique, à mon sens, sont les suivantes.

- Une perspective est la suppression de la topologie en arbre dans Macari. Cette perspective est déjà entamée sur certains points : à l'heure actuelle, la synchronisation se fait selon une structure de graphe orienté sans cycle (un nœud ne se synchronise pas forcément avec son père), et l'acheminement des trames ne suit pas toujours la topologie en arbre (comme cela sera présenté dans 3.1.3). En revanche, il reste des points à traiter concernant la tolérance aux pannes : la panne d'un coordinateur remet en cause les adresses de nombreux nœuds de l'arbre, et la panne du CPAN est complexe à gérer.
- Une autre perspective est la poursuite des réflexions autour des mécanismes de synchronisation, notamment décentralisés ou distribués. Le mécanisme de synchronisation actuel, basé sur la liste des coordinateurs, a en effet plusieurs inconvénients : il nécessite un temps qui dépend du nombre de coordinateurs, les balises ont une taille proportionnelle au nombre de coordinateurs, et la taille du cycle est limitée à environ 4 s (au-delà de 4 s, les imprécisions temporelles dues à la dérive des nœuds dépassent les intervalles de garde de Macari). Il serait donc intéressant de considérer d'autres mécanismes de synchronisation, basés sur des échanges locaux par exemple, et sans que le CPAN n'ait un rôle prépondérant. Il serait aussi intéressant d'étudier des mécanismes de synchronisation tolérants aux pannes (comme l'algorithme proposé dans [GBO⁺10]).

Une autre perspective consiste à intégrer une approche multi-canal à plusieurs niveaux dans Macari : pour la construction de la topologie en arbre, afin de réduire les collisions (comme proposé dans [ATRD12]) ou d'accélérer l'association (comme proposé dans [RPA⁺13]), ou pour faire de la réutilisation spatiale dans $[T_0; T_1]$ et dans $[T_1; T_2]$ (et non pas uniquement dans $[T_2; T_3]$).

3.1.3 Couche réseau d'Ocari

La couche réseau d'Ocari est constituée de deux protocoles de routage : le protocole Rocari pour les activités ordonnancées, et le protocole EOLSR pour les activités non ordonnancées. Dans cette partie, nous allons décrire ces deux protocoles.

3.1.3.1 Le protocole Rocari

Le protocole Rocari (signifiant Routage pour Ocari) [Cha09] a été développé par le LIMOS. Il est basé sur le routage hiérarchique de ZigBee, avec deux modifications. La première modification concerne les coordinateurs pour lesquels la destination finale est une feuille d'un coordinateur voisin (ou le coordinateur voisin lui-même). La trame est alors envoyée directement à ce coordinateur voisin. La deuxième modification concerne les coordinateurs pour lesquels la destination finale est dans la descendance d'un coordinateur voisin. La trame est alors envoyée directement à ce coordinateur voisin.

Rocari peut être utilisé dans l'intervalle $[T_1; T_2]$ de Macari. Selon la première modification, le chemin est raccourci dans le voisinage de la destination. Selon la deuxième modification, deux cas peuvent se produire.

- Si la feuille f_y que l'on cherche à atteindre n'est pas dans la descendance du coordinateur chargé de l'acheminement et que le deuxième cas particulier s'applique, alors le coordinateur chargé de l'acheminement envoie la trame à un voisin v qui est un ancêtre de la feuille f_y . Sans Rocari, la trame serait remontée jusqu'au premier ancêtre commun entre le coordinateur chargé de l'acheminement et f_y , puis serait redescendue en passant par v .
- Si la feuille f_y que l'on cherche à atteindre est dans la descendance du coordinateur chargé de l'acheminement et que le deuxième cas particulier s'applique, alors le coordinateur chargé de l'acheminement envoie la trame à un voisin v qui est plus proche de f_y que le coordinateur chargé de l'acheminement. Sans Rocari, la trame serait redescendue en passant par v .

L'avantage de Rocari est qu'il permet de raccourcir la longueur des chemins par rapport au routage hiérarchique, tout en conservant la borne sur le délai de bout en bout de quatre cycles. Rocari possède toutefois deux inconvénients. D'une part, Rocari nécessite que tous les coordinateurs voisins d'un nœud coordinateur soient actifs en même temps que le nœud, ce qui augmente la consommation énergétique globale. D'autre part, Rocari est moins efficace (en terme de longueur de chemins) que le protocole de routage raccourci : Rocari ne considère que deux cas particuliers, tandis que le routage raccourci considère tous les cas possibles.

3.1.3.2 Le protocole EOLSR

Le protocole EOLSR (*Energy efficient OLSR*) [MM08a] a été développé par Inria. EOLSR est un protocole de routage basé sur OLSR (décrit dans la partie 2.1.2), et met en place les fonctionnalités suivantes.

- La consommation énergétique des nœuds est prise en compte en calculant l'énergie dépensée en transmission, en réception et en écoute, pour chacun des nœuds.
- Le calcul des routes est basé sur l'énergie, grâce à une fonction de coût qui intègre la consommation des nœuds.
- Le choix des MPR est basé sur l'énergie, puisque ce choix est fait en fonction d'une estimation de la consommation énergétique du nœud, et de l'énergie résiduelle du nœud et de son voisinage à deux sauts.

EOLSR est utilisé dans la période $[T_2; T_3]$ pour acheminer les paquets. Il permet à tous les coordinateurs de communiquer en utilisant des routes courtes et économes en énergie, plutôt que de se baser sur la topologie logique arborescente (comme c'est le cas pour Rocari, utilisé dans la période $[T_1; T_2]$).

L'avantage principal d'EOLSR est sa faible consommation énergétique. En effet, lorsqu'un nœud transmet, seul le sous-ensemble de ses voisins qui sont ses MPR sont actifs. En revanche, EOLSR n'est pas basé sur un routage des plus courts chemins, et peut donc produire des routes longues (en terme de nombre de sauts).

3.1.3.3 Bilan de la couche réseau d'Ocari

Les travaux autour de la couche réseau d'Ocari se sont concentrés sur le protocole de routage Rocari, principalement parce que le développement d'un protocole de routage pour Ocari était la tâche du partenaire Inria. Nous avons pu collaborer en étudiant les interactions entre la couche liaison de données et la couche réseau (notamment, en proposant le protocole TDMA/CA).

Plus généralement, nous avons étudié les protocoles de la couche réseau dans les réseaux industriels au travers des performances des mécanismes d'association au réseau et d'adressage. Pendant la thèse de Nassima HADID, nous avons proposé un protocole d'association plus efficace que celui de IEEE 802.15.4 [HGM10*]. Pendant la thèse de Nancy EL RACHKIDY, nous avons travaillé sur l'étude des mécanismes d'adressage [ERGM09*, ERGM12c*]. Pendant le stage de master recherche de Michael ABDALLAH, nous avons évalué l'impact de la modélisation du canal et des règles d'associations sur l'arbre généré.

Une perspective à court terme est de développer un protocole de routage basé sur des adresses hiérarchiques (afin de pouvoir profiter des avantages du routage hiérarchique), mais pouvant supporter la réassociation de certains nœuds. Actuellement, la réassociation d'un nœud entraîne son changement d'adresse logique, et interrompt toutes les communications en cours avec l'ancienne adresse logique. Une solution étudiée dans l'équipe [MCHM13] consiste à demander à l'ancien coordinateur de rattachement de rediriger le trafic vers le nouveau coordinateur de rattachement, mais cela induit de longs délais. Une autre solution serait de pouvoir marquer certaines adresses dans les tables d'acheminement des coordinateurs, afin de pouvoir différencier les adresses hiérarchiques réelles et les adresses de nœuds réassociés.

3.1.4 Synthèse des protocoles réseau et MAC dans Ocari

Macari gère l'accès au médium de plusieurs manières différentes, comme résumé sur la figure 3.14. Dans $[T_0; T_1]$, les balises sont envoyées en TDMA (en respectant l'ordre de synchronisation). Dans $[T_1; T_2]$, le temps est divisé en plusieurs intervalles (l'exemple est donné pour trois coordinateurs). Au début de chacun de ces intervalles, CSMA/CA slotté est utilisé pour communiquer entre le coordinateur et ses feuilles. Ensuite, d'éventuels GTS sont utilisés, ce qui correspond à un accès TDMA. Finalement, le coordinateur dialogue avec son père selon un accès TDMA. Dans $[T_2; T_3]$, le temps est divisé en deux sous-périodes. La sous-période colorée est elle-même divisée en intervalles de temps (l'exemple est donné pour deux couleurs). Pendant

chacun de ces intervalles, TDMA/CA est utilisé. La sous-période non colorée utilise CSMA/CA slotté.

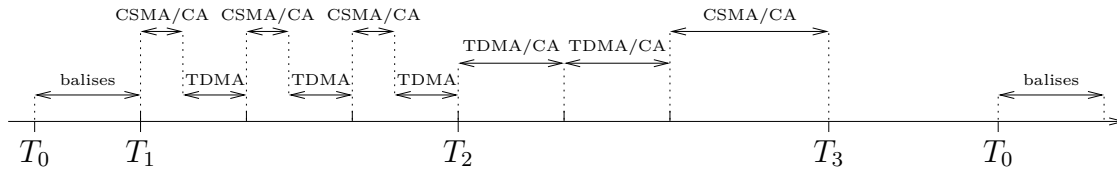


Figure 3.14 – L'accès au médium dans Macari varie en fonction du temps : il peut s'agir d'un accès TDMA, d'un accès de type CSMA/CA slotté, ou d'un accès de type TDMA/CA.

L'architecture d'Ocari est donc une architecture multi-piles : l'une des piles protocolaires concerne le trafic contraint, et l'autre pile concerne le trafic non contraint. Ces deux piles se rejoignent au niveau de la sous-couche MAC (voir figure 3.1). Au niveau réseau, les trames correspondant au trafic contraint sont prises en charge par le protocole Rocari (ces trames sont acheminées pendant la période $[T_1; T_2]$ lors de la collecte avec contention, ainsi qu'au moyen de GTS et de communications père-fils), et les trames correspondant au trafic non contraint sont prises en charge par le protocole EOLSR (ces trames sont acheminées pendant la période $[T_1; T_2]$ lors de la collecte avec contention, et pendant la période $[T_2; T_3]$).

3.2 Architectures multi-piles simples et locales

Plusieurs architectures réseaux existantes (comme l'architecture réseau définie dans le cadre du projet Ocari) peuvent être représentées par des architectures utilisant plusieurs piles protocolaires, comme nous l'avons montré pendant la thèse de Nancy EL RACHKIDY [ERGM11*]. Dans cette partie, nous allons introduire deux types d'architectures multi-piles : les architectures simples et les architectures locales.

3.2.1 Description des architectures simples

Définition 2 (Architecture simple). Une architecture multi-piles est dite simple quand, pour chaque paquet, une seule pile protocolaire est utilisée pour l'acheminement de bout en bout.

Le concept d'architecture simple repose sur un concept que nous appelons *multi-layering* [ERGM11*], qui permet à plusieurs protocoles différents de coexister au sein d'une même couche. Ce concept est différent de celui du *cross-layering*, qui permet à des protocoles de couches non adjacentes de communiquer. *Multi-layering* et *cross-layering* sont complémentaires, le *multi-layering* permettant d'ailleurs d'augmenter les possibilités d'interactions entre protocoles (et donc le *cross-layering*), puisqu'il augmente le nombre de protocoles impliqués.

Dans la description qui suit, nous ferons deux hypothèses pour simplifier la discussion. Tout d'abord, nous supposerons que les piles protocolaires ont des protocoles

MAC différents (ce qui est le cas dans Ocari)²³ : chaque pile protocolaire i est ainsi caractérisée par un ensemble²⁴ de protocoles applicatifs \mathcal{A}_i , un protocole réseau \mathcal{R}_i et un protocole d'accès au médium \mathcal{M}_i . Ensuite, nous supposons que les piles protocolaires suivent un calendrier d'activation, ce qui permet à chaque protocole d'accès au médium de gérer le temps pendant un intervalle dédié.

3.2.1.1 Calendrier d'activation des piles

Une architecture simple utilisant n piles protocolaires et ayant des couches MAC différentes est basée sur un calendrier composé de n périodes p_i qui se répètent (généralement cycliquement). Pendant chaque période p_i , tous les nœuds du réseau utilisent la combinaison, notée $(\mathcal{R}_i, \mathcal{M}_i)$, d'un protocole de routage \mathcal{R}_i et d'un protocole MAC \mathcal{M}_i . La cohérence des visions du temps de chacun des nœuds est garantie par un mécanisme de synchronisation. Nous supposons que ce mécanisme de synchronisation, exécuté par tous les nœuds, se répète lui aussi (cela nous permet de considérer que la période de synchronisation est la première période du calendrier). Un protocole MAC commun à toutes les piles, \mathcal{M}^r , est chargé de réaliser cette synchronisation entre tous les nœuds, et de s'assurer que les nœuds connaissent l'instant de début du calendrier²⁵. \mathcal{M}^r est aussi chargé de construire et de communiquer à tous les nœuds le calendrier, ce qui inclut l'identification de chaque période p_i , leur instant de début et leur durée. Finalement, \mathcal{M}^r est chargé de faire la liaison entre tous les protocoles \mathcal{M}_i et la couche physique (notamment, \mathcal{M}^r redirige les trames reçues de la couche physique au protocole \mathcal{M}_i adéquat, par exemple en fonction de leur instant de réception). Il peut exister des périodes d'inactivité pour tous les nœuds dans le calendrier. Certains nœuds peuvent aussi dormir pendant certaines périodes p_i , sous la responsabilité du protocole \mathcal{M}_i correspondant.

La figure 3.15 présente un exemple d'architecture simple, utilisant trois piles protocolaires. Chaque pile protocolaire i dispose de ses applications, de son protocole de routage, et de son protocole MAC. Le protocole MAC commun \mathcal{M}^r permet la coexistence des piles, et le respect du calendrier d'activation des piles.

La figure 3.16 présente un calendrier pour $n = 3$ périodes (sans compter la période de synchronisation). Sur cet exemple, on peut remarquer que les périodes p_1 , p_2 et p_3 n'ont pas la même durée.

La construction du calendrier (réalisée par un nœud ayant des ressources importantes, comme le CPAN) peut prendre plusieurs paramètres en compte : le choix des protocoles MAC et réseau en fonction de leurs propriétés, la durée allouée à chaque

23. Des exemples d'architectures simples où les piles protocolaires ont des protocoles MAC identiques sont mentionnés dans 3.2.1.2.

24. On pourrait se dire que la coexistence de plusieurs protocoles par couche existe au niveau applicatif dans les piles protocolaires traditionnelles. Toutefois, la différence entre la coexistence des protocoles applicatifs et la coexistence des protocoles des autres couches est importante. Chaque protocole applicatif fournit un service différent, tandis que les protocoles des couches inférieures fournissent des services définis lors du découpage en couches, et ces services nécessitent des responsabilités qui ne sont pas supposées être partagées (comme la gestion du routage, du temps ou du médium).

25. Nous ne détaillons pas le mécanisme de synchronisation utilisé, qui peut s'appuyer sur plusieurs protocoles existants, comme RBS [EGE02], TPSN [GKS03], l'intervalle $[T_0; T_1]$ de Macari, ADCF [LvdBC11] ou SISP [vdBVD11].

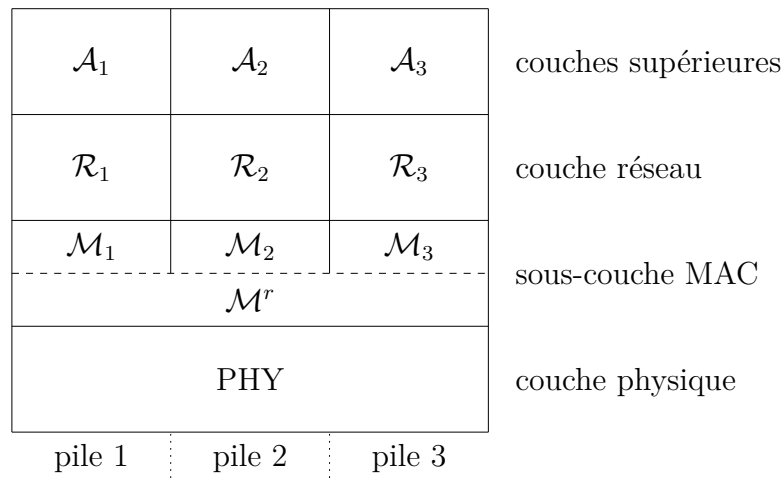


Figure 3.15 – Architecture multi-piles simple. La réunion des trois piles se fait par le protocole \mathcal{M}^r au niveau de la sous-couche MAC.

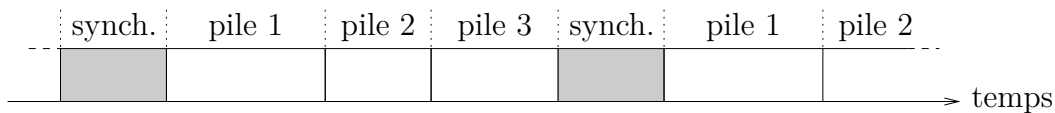


Figure 3.16 – Calendrier dans lequel se répètent $n = 3$ périodes, correspondant à trois piles protocolaires différentes.

combinaison de protocoles, la quantité estimée de trafic par période et les propriétés de ce trafic, ou encore les besoins de synchronisation des protocoles MAC (les protocoles nécessitant une grande précision de synchronisation doivent être placés tôt dans le calendrier). Il est important de remarquer que le choix des combinaisons de protocoles et la durée allouée à chaque combinaison ont un impact important sur les performances de la solution finale.

3.2.1.2 Exemples d'architectures multi-piles simples

Multi-piles ayant des protocoles MAC différents.

IEEE 802.15.4 en mode suivi de balises peut être considéré comme une architecture multi-piles ayant des protocoles MAC différents. Pour s'en convaincre, on peut noter que IEEE 802.15.4 en mode suivi de balises utilise un calendrier cadencé par l'envoi de balises (ces balises correspondent au protocole de synchronisation, intégré à \mathcal{M}_r). À la réception d'une balise ou à la fin de l'envoi d'une balise, les nœuds entrent dans la période CAP (dont la durée peut être calculée à partir des informations de la balise) pendant laquelle l'accès au médium est géré par CSMA/CA slotté (ce qui correspond au protocole MAC \mathcal{M}_1). Puis, les nœuds entrent dans la période CFP (dont la durée est connue et peut être variable) pendant laquelle l'accès au médium est géré par des GTS (ce qui correspond au protocole MAC \mathcal{M}_2).

Multi-piles ayant le même protocole MAC.

La pile duale IPv4-IPv6 [RFC 4213] (présentée dans la partie 2.2.2.1) est une architecture multi-piles dans laquelle la séparation se fait au niveau réseau (et donc, le même protocole MAC est utilisé par les deux piles). Une pile duale IPv4-IPv6 correspond à une implémentation conjointe des protocoles IPv4 (correspondant au protocole réseau \mathcal{R}_1) et IPv6 (correspondant au protocole réseau \mathcal{R}_2), ce qui permet d'éviter dans de nombreux cas les encapsulations d'IPv6 dans IPv4 ou d'IPv4 dans IPv6. Lorsqu'un paquet est reçu par la couche réseau, l'entête de ce paquet est examiné : si le champ *version* vaut 4, le paquet est transmis à la pile IPv4, et si ce champ vaut 6, le paquet est transmis à la pile IPv6.

Dans [MHJ⁺12] (décrit dans la partie 2.2.2.3), les auteurs proposent une autre architecture multi-piles dans laquelle la séparation se fait au niveau réseau. Ils détaillent l'utilisation d'une stratégie de routage nommée R_{e+d} qui permet de combiner la réduction de l'énergie consommée et du délai. Deux types de paquets sont générés au niveau applicatif : les paquets urgents sont acheminés selon la stratégie de routage R_d (correspondant au protocole réseau \mathcal{R}_1) qui réduit le délai, et les paquets non urgents sont acheminés selon la stratégie de routage R_e (correspondant au protocole réseau \mathcal{R}_2) qui réduit l'énergie consommée.

3.2.1.3 Problèmes des architectures simples

Dans les architectures simples, c'est l'application du nœud source qui détermine la pile qui est utilisée pour acheminer chaque paquet. L'acheminement des paquets est donc imposé par un nœud qui a une vision limitée du réseau, ce qui peut être pénalisant.

Dans les architectures simples ayant des protocoles MAC différents, qui sont donc basées sur un calendrier, des problèmes supplémentaires apparaissent. Le premier problème est le dimensionnement de chaque période du calendrier, qui a un impact important sur les performances de l'architecture. En effet, quand une période est sous-dimensionnée, certaines trames doivent attendre la prochaine période pour être traitées, ce qui augmente leur délai, et quand une période est sur-dimensionnée, du temps est inutilement perdu. Le second problème est le délai induit par les périodes : une trame ne peut pas être traitée hors de la période pour laquelle elle est marquée, même si cette trame est prioritaire.

3.2.2 Description des architectures locales

Définition 3 (Architecture locale). Une architecture multi-piles est dite locale quand, pour chaque paquet, plusieurs piles protocolaires peuvent participer à l'acheminement de bout en bout, et quand la décision d'utiliser une pile ou une autre dépend du contexte local de chaque nœud du chemin. Le contexte local peut inclure, par exemple, le temps, l'identifiant du nœud, l'énergie résiduelle dans le nœud, ou les statistiques collectées par le nœud.

3.2.2.1 Échanges de paquets entre files d'attente

Pour réduire l'impact des problèmes des architectures simples ayant des protocoles MAC différents (à savoir, le manque de flexibilité de la pile utilisée, le dimensionnement difficile des périodes du calendrier, et le délai induit par ces périodes), nous présentons le mécanisme d'échanges de paquets entre files d'attente proposé pour Ocari dans [ERCGM11*]. Avec ce mécanisme d'échanges, une trame prévue pour une période p_i peut être traitée pendant une période p_j (avec $i \neq j$), par exemple lorsque la file d'attente \mathcal{Q}_j des trames pour p_j est vide (pour ne pas pénaliser la période p_j)²⁶. Il faut noter que le nœud chargé d'acheminer la trame n'est pas toujours en mesure de communiquer la trame au nœud correspondant au prochain saut calculé selon \mathcal{R}_i , car le protocole \mathcal{M}_j a pu décider de l'inactivité (ou du changement de fréquence) de ce nœud. Si le nœud correspondant au prochain saut calculé selon \mathcal{R}_i est actif pour \mathcal{M}_j , la trame peut être envoyée. Sinon, il faut que le prochain saut de la trame soit recalculé selon \mathcal{R}_j avant d'être envoyé par \mathcal{M}_j .

La figure 3.17 présente les trajets possibles (représentés par des flèches épaisses) pour une trame envoyée par une application de \mathcal{A}_1 , dans une architecture multi-piles. Les files d'attente (\mathcal{Q}_1 et \mathcal{Q}_2) ont été représentées sur le schéma. Sans échange de files d'attente (à gauche sur la figure), le prochain saut de la trame est calculé selon \mathcal{R}_1 , et la trame est acheminée selon \mathcal{M}_1 . Avec l'échange de files d'attente (à droite sur la figure), trois cas peuvent se produire :

- si c'est le protocole \mathcal{M}_1 qui est actif, la trame est acheminée selon \mathcal{M}_1 (étant donné que le prochain saut de la trame, calculé selon \mathcal{R}_1 , est obligatoirement actif),
- si c'est le protocole \mathcal{M}_2 qui est actif, et si le prochain saut selon \mathcal{R}_1 est joignable pendant \mathcal{M}_2 , la trame est acheminée selon \mathcal{M}_2 ,
- si c'est le protocole \mathcal{M}_2 qui est actif, et si le prochain saut selon \mathcal{R}_1 n'est pas joignable pendant \mathcal{M}_2 , le prochain saut est recalculé selon \mathcal{R}_2 et la trame est acheminée selon \mathcal{M}_2 .

Pour que l'échange de paquets puisse être implémenté, une solution est de marquer les trames par le numéro de la période choisie par l'application source. Quand une trame t , marquée pour p_i , est considérée pour une période p_j (avec $i \neq j$), le marquage de la trame n'est pas changé. Si t n'a pas pu être envoyée à la fin de la période p_j , t retourne dans sa file d'attente initiale \mathcal{Q}_i . t peut être reconsidérée pour une transmission dans une autre période, incluant la prochaine période p_i . Le compteur de retransmissions n'est augmenté pour t que lors de la période p_i (afin de ne pas pénaliser les trames qui sont considérées pendant d'autres périodes).

Ce mécanisme d'échanges de paquets entre files d'attente permet de résoudre les problèmes décrits dans 3.2.1.3. En effet, le sur-dimensionnement des périodes n'est plus un inconvénient majeur, car le temps inutilisé pour une période p_j peut être réutilisé pour les trames des autres périodes. Le sous-dimensionnement n'est plus critique non plus, puisqu'une trame marquée pour p_i peut être envoyée dans une période p_j (avec $i \neq j$), sans avoir à attendre la prochaine occurrence de la période p_i . Le surcoût de l'échange de paquets entre files d'attente vient du marquage de

26. Plus généralement, on peut concevoir un mécanisme qui détermine quelle est la trame la plus prioritaire dans toutes les files d'attente, en fonction de la période actuelle.

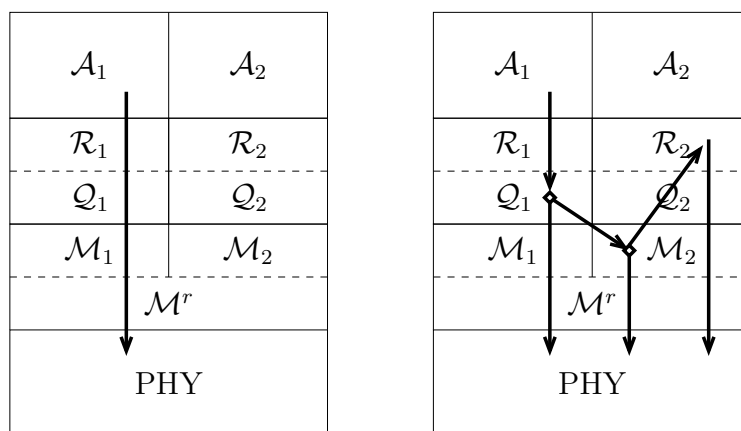


Figure 3.17 – Trajets possibles pour une trame marquée pour p_1 , dans une architecture sans échange entre files d’attente (à gauche) et avec échanges entre files d’attente (à droite). Quand les échanges entre files d’attente sont permis, la trame peut être manipulée par plusieurs piles protocolaires.

chaque trame (supposé réduit) et de l’utilisation plus importante du processeur, notamment pendant des périodes durant lesquelles il aurait pu être inactif.

3.2.2.2 Autres exemples d’architectures locales

Un réseau dans lequel deux versions d’un protocole de routage coexistent peut être considéré comme une architecture multi-piles locale. Notons \mathcal{R}_1 la version initiale d’un protocole de routage, et \mathcal{R}_2 une version plus récente de ce protocole. Les paquets circulant sur le réseau sont acheminés tantôt par \mathcal{R}_1 , tantôt par \mathcal{R}_2 , selon les nœuds rencontrés sur le chemin. Cette situation peut se produire lors du déploiement incrémental d’un protocole, c’est-à-dire lorsqu’il n’est pas possible d’effectuer la mise à jour de tous les nœuds simultanément. Cet exemple correspond bien à la définition d’une architecture locale²⁷, où le contexte local est lié à la version du protocole installé.

Les communications en aparté, que nous avons étudiées pendant la thèse de Nassima HADID [Had11], constituent un autre exemple d’architecture multi-piles locale. Dans le contexte des communications en aparté, un réseau est déployé sur des entités mobiles [HGM10*, HGM11a*, HGM11b*]. Le réseau de chaque mobile fonctionne sur un canal propre à ce mobile. Chaque réseau opère selon deux modes. Dans le premier mode, les nœuds suivent le protocole IEEE 802.15.4 en mode suivi de balises, avec ses périodes d’activité et ses périodes d’inactivité. Dans le deuxième mode, qui a lieu pendant la période d’inactivité du premier mode, les nœuds riches en énergie de tous les mobiles basculent sur un canal commun, dit canal d’aparté, et sont capables de communiquer entre eux lorsqu’ils sont à portée. L’algorithme

27. Dans cet exemple et à ce point du document, nous considérons que la version du protocole de chaque nœud est fixée et n’évolue pas dans le temps. La prise en compte complète du déploiement incrémental d’un protocole, c’est-à-dire du moment où tous les nœuds sont dans la version initiale du protocole de routage, au moment où tous les nœuds sont dans la version la plus récente du protocole, correspond en fait à une architecture globale. Les architectures globales sont étudiées dans la partie 4.2 de ce document.

CSMA/CA non slotté est utilisé pendant ce deuxième mode, et un protocole de routage tolérant au délai est utilisé pour que les nœuds de mobiles différents puissent communiquer entre eux. La différence entre les deux protocoles de routage utilisés est importante, puisque l'un des protocoles de routage est interne à chaque mobile, alors que l'autre peut être externe. Dans cette architecture, c'est le temps et l'identité du mobile destination qui déterminent la pile protocolaire à utiliser.

3.2.2.3 Apparition de boucles

Des boucles de routage peuvent apparaître lorsque des échanges entre files d'attente sont permis, puisqu'il est alors possible qu'une même trame soit routée selon plusieurs protocoles de routage différents [ERCGM11*].

La figure 3.18 présente un exemple de réseau de quatre nœuds (a , b , c et d), avec deux protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 . Le protocole de routage \mathcal{R}_1 pour la destination d est représenté par des lignes continues avec des flèches pleines, et le protocole de routage \mathcal{R}_2 pour la destination d par des lignes pointillées avec des flèches creuses. On peut remarquer que \mathcal{R}_1 seul (respectivement \mathcal{R}_2 seul) ne conduit pas à l'apparition de boucles, ce qui est attendu pour un protocole de routage consistant. En d'autres termes, si tous les nœuds du réseau envoient les paquets pour d selon \mathcal{R}_1 seul (respectivement \mathcal{R}_2 seul), aucune boucle ne se produit. Cependant, si certains nœuds du réseau envoient les paquets selon \mathcal{R}_1 , et d'autres selon \mathcal{R}_2 , des boucles potentiellement infinies peuvent se produire. C'est le cas si b décide d'envoyer systématiquement les paquets pour d selon \mathcal{R}_1 , alors que c décide d'envoyer systématiquement les paquets pour d selon \mathcal{R}_2 .

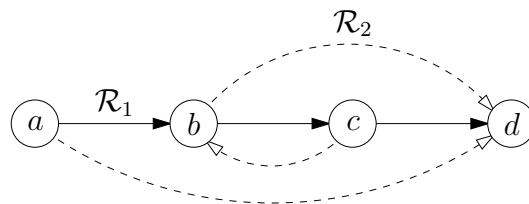


Figure 3.18 – L'utilisation conjointe des protocoles \mathcal{R}_1 et \mathcal{R}_2 peut provoquer des boucles infinies de routage, quand le nœud b décide d'envoyer les paquets pour d selon \mathcal{R}_1 , et que le nœud c décide de les envoyer selon \mathcal{R}_2 .

La figure 3.19 présente un exemple de réseau de cinq nœuds (a , b , c , d et e), avec deux protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 , pour la destination e . Comme précédemment, on peut remarquer que \mathcal{R}_1 seul (respectivement \mathcal{R}_2 seul) ne conduit pas à l'apparition de boucles. Suivons un trajet possible d'un paquet partant du nœud a et à destination du nœud e , en supposant que la décision d'utiliser \mathcal{R}_1 ou \mathcal{R}_2 change tous les deux sauts et que \mathcal{R}_1 est le premier protocole à être utilisé. Le paquet est acheminé selon \mathcal{R}_1 de a en b puis en d . Ensuite, le paquet est acheminé selon \mathcal{R}_2 de d en b puis en c . Finalement, le paquet est acheminé selon \mathcal{R}_1 de c en d puis en e . Le chemin complet du paquet a été (a, b, d, b, c, d, e) , ce qui correspond à six sauts. À cause d'une boucle finie entre b et d , le paquet a parcouru un long chemin²⁸.

28. La longueur du chemin parcouru est supérieure à la longueur du chemin selon \mathcal{R}_1 seul, ou selon \mathcal{R}_2 seul.

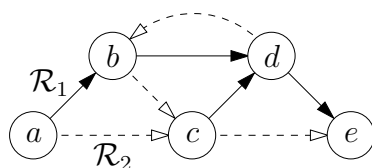


Figure 3.19 – L’utilisation conjointe des protocoles \mathcal{R}_1 et \mathcal{R}_2 peut provoquer des boucles finies de routage.

Les boucles de routage constituent un phénomène à éviter. En effet, elles conduisent à une baisse globale des performances du réseau, due à une augmentation du délai, à une augmentation de la taille des files d’attente, et à une consommation inutile de la bande passante.

3.3 Évitement des boucles entre protocoles de routage

Dans cette partie, nous étudions comment les boucles causées par les architectures multi-piles locales peuvent être évitées, ou comment leurs effets peuvent être réduits. Ces travaux ont été effectués principalement dans le cadre de la thèse de Nancy EL RACHKIDY [ER11].

L’apparition de boucles de routage dépend à la fois des protocoles de routage (puisque ils décident des routes) et des protocoles MAC (puisque c’est eux qui prennent la décision d’acheminer des trames ou non). Les boucles de routage infinies ont peu de chances d’apparaître en pratique, car elles nécessitent des alternances périodiques de protocoles de routage. Ces alternances périodiques sont globalement peu probables, grâce notamment aux protocoles MAC, pour les raisons suivantes.

- Le caractère aléatoire de nombreuses méthodes d’accès réduit le caractère systématique de la répétition.
- Le temps passé par chaque nœud en file d’attente est variable (car d’autres paquets se trouvent probablement dans la file d’attente de chaque nœud) : même si les protocoles de routage sont alternés périodiquement, cette répétition a peu de chances de concerner le même paquet.
- En présence de boucles, le réseau devient surchargé, et les paquets impliqués dans les boucles ont des chances d’être supprimés par les protocoles MAC (lors du dépassement des compteurs de retransmission).

Nous nous intéressons cependant aux propositions de routage permettant de réduire ces boucles, car l’évitement des boucles est de la responsabilité de la couche réseau, et non pas de la sous-couche MAC. De plus, même si les trames sont détruites par les protocoles MAC en cas de congestion du réseau, elles le sont indépendamment du fait qu’elles soient la cause de boucles. Finalement, la présence de boucles, même en proportion faible, peut réduire significativement les performances globales du réseau.

Dans la suite, nous allons étudier des propriétés que l’on peut donner à des

ensembles de protocoles de routage²⁹. Nous allons montrer comment ces propriétés permettent d'éviter (ou de limiter) les boucles liées à l'utilisation de ces ensembles de protocoles.

3.3.1 Protocoles compatibles

Définition 4 (Protocole de routage). Soit $G = (V, E)$ un graphe orienté, avec V un ensemble de nœuds et E un ensemble d'arêtes. Un protocole de routage \mathcal{R} est défini comme une fonction de $V \times V \rightarrow V$, telle que $\mathcal{R}(n, d)$ est le prochain saut de n à destination de d (avec $(n, \mathcal{R}(n, d)) \in E$). Cette définition fait l'hypothèse que $\mathcal{R}(d, d) = d$ pour tout $d \in V$.

Définition 5 (Boucle de routage infinie). Soit p le chemin que suit un paquet pour une destination d . p entre dans une boucle de routage infinie si p est de longueur infinie et ne contient pas d .

Définition 6 (Boucle de routage finie). Soit p le chemin que suit un paquet pour une destination d . Le paquet entre dans une boucle de routage (finie) si le chemin p passe deux fois (ou plus) par un même nœud (à l'exception de d).

Définition 7 (Protocoles de routage compatibles [ERGM11*]). Deux protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 sont dits compatibles si et seulement si un nœud quelconque peut décider arbitrairement de router chaque paquet selon \mathcal{R}_1 ou \mathcal{R}_2 , sans que les paquets n'entrent dans une boucle.

La figure 3.20 présente un exemple de deux protocoles de routage compatibles (pour une destination d). On remarque qu'il est impossible que les paquets entrent dans une boucle de routage, quelque soit la décision de routage prise par chacun des nœuds.

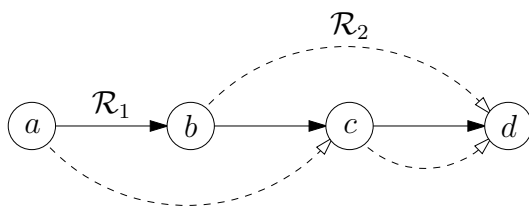


Figure 3.20 – Les protocoles \mathcal{R}_1 et \mathcal{R}_2 sont compatibles (pour la destination d) : tout nœud peut décider arbitrairement d'envoyer selon \mathcal{R}_1 ou selon \mathcal{R}_2 , sans causer de boucle.

3.3.1.1 Caractérisation des protocoles compatibles

Définition 8 (Fonction de décision f_d). Soit un graphe orienté $G = (V, E)$, une destination $d \in V$, et un ensemble de protocoles de routage $\{\mathcal{R}_i\}_i$. Une fonction de

²⁹. Dans la plupart des travaux de recherche sur les protocoles de routage, les chercheurs s'intéressent principalement aux propriétés de protocoles de routage donnés. Ici, nous nous concentrons davantage sur les propriétés des ensembles de protocoles, sans présumer des propriétés des protocoles constituant cet ensemble.

décision f_d est une fonction de $V \rightarrow \mathbb{N}$ telle que pour tout nœud n et pour tout i , n est autorisé à transmettre un paquet pour une destination d par le prochain saut $\mathcal{R}_i(n, d)$ si et seulement si $f_d(\mathcal{R}_i(n, d)) < f_d(n)$.

Théorème 1. *Soit $G = (V, E)$ un graphe orienté, \mathcal{R}_1 et \mathcal{R}_2 deux protocoles de routage, et $d \in V$ une destination. S'il existe une fonction de décision f_d telle que $f_d(d) = 0$ et $\forall n \in V \setminus \{d\}$, $\max\{f_d(\mathcal{R}_1(n, d)), f_d(\mathcal{R}_2(n, d))\} < f_d(n)$, alors \mathcal{R}_1 et \mathcal{R}_2 sont compatibles [ER11, ERGM13a*].*

Démonstration. Soit $G = (V, E)$ un graphe orienté, \mathcal{R}_1 et \mathcal{R}_2 deux protocoles de routage, et $d \in V$ une destination. Nous supposons qu'il existe une fonction de décision f_d telle que $f_d(d) = 0$ et $\forall n \in V \setminus \{d\}$, $\max\{f_d(\mathcal{R}_1(n, d)), f_d(\mathcal{R}_2(n, d))\} < f_d(n)$. Nous cherchons à montrer que tout chemin partant d'un nœud arbitraire n et à destination de d , finit par atteindre d en un nombre fini de sauts, sans générer de boucle. En d'autres termes, nous cherchons à montrer que pour toute séquence $(r_i)_i$ de décisions de routage (avec $r_i \in \{1, 2\}$ pour tout i), si $(r_i)_i$ est suffisamment grande, le chemin qui part de n et qui suit les décisions de routage $(r_i)_i$ atteint d en un nombre fini de sauts, sans générer de boucle.

Posons $(r_i)_i$ une suite arbitraire (mais suffisamment longue) de décisions de routage. Définissons le chemin p comme $p = (n_0, n_1, n_2, \dots)$, tel que $n_0 = n$ et $n_{i+1} = \mathcal{R}_{r_i}(n_i, d)$. La construction de p s'arrête uniquement quand d est atteint. Par construction, p est unique.

Tout d'abord, nous allons montrer que si p est infini, tous les nœuds de p appartiennent à $V \setminus \{d\}$. Supposons donc que p est infini. À chaque nœud n_i de p correspond un nœud n_{i+1} de p , ce qui est possible uniquement si $n_i \neq d$ (par construction). Ainsi, tous les nœuds de p sont dans $V \setminus \{d\}$.

Montrons à présent que p est fini. Raisonnons par contradiction en supposant que p est infini. Puisque p est infini, tous les nœuds de p sont différents de d . On peut construire la suite $s = (f_d(n_0), f_d(n_1), f_d(n_2), \dots)$ en appliquant la fonction f_d à chaque nœud de p . Or, pour chaque nœud $n_i \neq d$, $f_d(n_{i+1}) = f_d(\mathcal{R}_{r_i}(n_i, d)) < f_d(n_i)$ par définition de f_d . La suite s est donc strictement décroissante. f_d prenant ses valeurs dans \mathbb{N} , il en va de même pour s . Il n'est pas possible d'avoir une suite infinie de valeurs strictement décroissante dans \mathbb{N} : l'hypothèse que p est infini était fautive, donc p est fini.

Nous allons montrer que tous les nœuds de p (à l'exception du dernier), appartiennent à $V \setminus \{d\}$, et que le dernier nœud de p est d . Nous savons que p est fini. p peut donc s'écrire $(n_0, n_1, n_2, \dots, n_k)$. Par construction, pour tout $i \in [0; k-1]$, n_{i+1} existe, donc $n_i \neq d$. $n_k = d$ puisque la construction de p s'est arrêtée (cela ne peut pas provenir de l'arrêt de la séquence $(r_i)_i$, puisqu'elle a été choisie suffisamment longue). On a bien la propriété attendue.

Il nous reste à présent à montrer que p ne contient pas de boucle. Par contradiction, supposons que p passe deux fois par un même nœud. Il existe donc x et y tels que $n_x = n_y$, avec $x < y$. Nous avons deux cas possibles : soit $y = k$, soit $y \neq k$.

- Supposons $y = k$. Cela signifie que $n_y = n_k = d$, et donc que $n_x = d$. Toutefois, n_x admet un successeur dans p (puisque'il existe au moins un nœud n_y après n_x dans p), ce qui est impossible par construction de p .

- Supposons $y \neq k$. Donc, n_y n'est pas le dernier nœud de p , ce qui signifie que tous les nœuds de n_0 à n_y (au moins) appartiennent à $V \setminus \{d\}$. Pour chaque nœud $n_i \in (n_0, n_1, n_2, \dots, n_x, \dots, n_y)$, nous avons $f_d(n_{i+1}) = f_d(\mathcal{R}_{r_i}(n_i, d)) < f_d(n_i)$. On a notamment $f_d(n_x) < f_d(n_y)$ car $x < y$. Or, $n_x = n_y$, donc $f_d(n_x) = f_d(n_y)$, ce qui est impossible.

Dans les deux cas, nous avons une contradiction. p ne passe donc pas deux fois par le même nœud.

Nous avons donc montré que l'unique chemin p atteint d en un nombre fini de sauts, sans passer deux fois par le même nœud, et indépendamment de la séquence de décisions $(r_i)_i$ des nœuds. \mathcal{R}_1 et \mathcal{R}_2 sont donc compatibles. \square

Cette preuve peut être décrite de manière plus intuitive, en définissant la fonction f_d comme une distance à la destination d . Comme la distance $f_d(\mathcal{R}_i(n, d))$ d'un nœud quelconque n à la destination d selon chaque protocole de routage \mathcal{R}_i , est inférieure à la distance $f_d(n)$ du nœud actuel n à d , la suite des valeurs de la fonction f_d pour les nœuds d'un chemin doit forcément décroître. Cette décroissance impose que le chemin soit sans boucle, et qu'il termine en d .

L'inconvénient majeur du théorème 1 est qu'il ne permet ni de construire une fonction f_d qui rend les protocoles compatibles, ni de prouver l'existence d'une telle fonction.

3.3.1.2 Exemples de protocoles compatibles

Propriété 7. *Les protocoles de routage des plus courts chemins sont compatibles entre eux.*

Démonstration. Notons \mathcal{R}_1 et \mathcal{R}_2 deux protocoles de routage des plus courts chemins. Construisons f_d de la manière suivante : pour tout nœud $n \in V$, $f_d(n)$ est le plus petit nombre de sauts de n à d .

Montrons que f_d satisfait aux conditions du théorème 1. Tout d'abord, f_d est bien une fonction définie de V dans \mathbb{N} et $f_d(d) = 0$ (car f_d est une distance). Ensuite, pour tous les nœuds $n \in V \setminus \{d\}$, nous avons $f_d(n) > f_d(\mathcal{R}_i(n, d))$ par définition de \mathcal{R}_i (pour tout i). Par conséquent, on a bien $f_d(n) > \max\{f_d(\mathcal{R}_1(n, d)), f_d(\mathcal{R}_2(n, d))\}$, ce qui signifie que les protocoles \mathcal{R}_1 et \mathcal{R}_2 sont compatibles. \square

Propriété 8. *Le protocole de routage hiérarchique et le protocole de routage raccourci sont compatibles.*

Démonstration. Notons \mathcal{R}_1 le protocole de routage hiérarchique et \mathcal{R}_2 le protocole de routage raccourci. Construisons f_d de la manière suivante : pour tout nœud $n \in V$, $f_d(n)$ est le nombre de sauts de n à d sur l'arbre.

Montrons que f_d satisfait aux conditions du théorème 1. Tout d'abord, f_d est bien une fonction définie de V dans \mathbb{N} et $f_d(d) = 0$ (car f_d est une distance).

- Pour tous les nœuds $n \in V \setminus \{d\}$, nous avons $f_d(n) > f_d(\mathcal{R}_1(n, d))$. En effet, \mathcal{R}_1 est le protocole de routage hiérarchique et f_d est la distance sur l'arbre, donc $f_d(n) = f_d(\mathcal{R}_1(n, d)) + 1$.
- Pour tous les nœuds $n \in V \setminus \{d\}$, nous avons $f_d(n) > f_d(\mathcal{R}_2(n, d))$. En effet, \mathcal{R}_2 est le protocole de routage raccourci, ce qui signifie que si x est le voisin

choisi par \mathcal{R}_2 , pour tout autre voisin v de n , on a $f_d(v) \geq f_d(x)$. Le prochain saut de n sur l'arbre, $\mathcal{R}_1(n, d)$, étant l'un des voisins de n , on a en particulier $f_d(\mathcal{R}_1(n, d)) \geq f_d(x)$. On a donc $f_d(n) > f_d(\mathcal{R}_1(n, d)) \geq f_d(x) = f_d(\mathcal{R}_2(n, d))$. On a donc $f_d(n) > \max\{f_d(\mathcal{R}_1(n, d)), f_d(\mathcal{R}_2(n, d))\}$, ce qui signifie que les protocoles \mathcal{R}_1 et \mathcal{R}_2 sont compatibles. \square

Il existe des protocoles de routage non compatibles, comme par exemple le protocole de routage hiérarchique et les protocoles de routage des plus courts chemins. On peut voir sur la figure 3.21 une topologie dans laquelle une boucle infinie peut se produire quand la source des paquets est d , que la destination des paquets est e , que b envoie selon un protocole de routage des plus courts chemins, et que d envoie selon le protocole de routage hiérarchique.

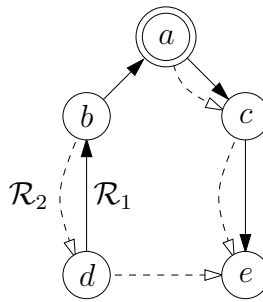


Figure 3.21 – Le protocole de routage hiérarchique et les protocoles de routage des plus courts chemins ne sont pas compatibles : une boucle peut se produire entre les nœuds b et d , quand la destination est e .

3.3.1.3 Protocoles ρ -compatibles

Dans cette partie, nous allons montrer que l'obtention d'une fonction de décision satisfaisant aux conditions du théorème 1 est un problème difficile dans le cas général (même lorsque les conditions du théorème 1 ne sont pas satisfaites pour un certain pourcentage de nœuds).

Définition 9 (Protocoles de routage ρ -compatibles). Soit un graphe orienté $G = (V, E)$, une destination $d \in V$, et deux protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 . Les protocoles \mathcal{R}_1 et \mathcal{R}_2 sont dits ρ -compatibles pour la destination d et pour la fonction f_d si et seulement si $f_d(n) > \max\{f_d(\mathcal{R}_1(n, d)), f_d(\mathcal{R}_2(n, d))\}$ pour un pourcentage ρ (au moins) des nœuds. Les nœuds qui ne vérifient pas la condition sont dits en conflit.

Si deux protocoles de routage sont 1-compatibles, ils sont compatibles. Si deux protocoles de routage sont ρ -compatibles avec $\rho < 1$, la fonction f_d trouvée ne permet pas de dire que les protocoles sont compatibles, puisque certains nœuds restent en conflit.

Définition 10 (Problème de détermination d'une fonction de décision (PDFD)). Soit un graphe orienté $G = (V, E)$, une destination $d \in V$, un pourcentage ρ , et deux protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 . Le problème PDFD consiste à déterminer l'existence d'une fonction de décision f_d telle que \mathcal{R}_1 et \mathcal{R}_2 sont ρ -compatibles.

Théorème 2. *PDFD est un problème NP-complet [ERGM13b*].*

Démonstration. Nous allons montrer que PDFD est NP-complet par réduction à partir du problème de couverture d'ensemble (*Set Cover Problem*, abrégé en SCP), qui est connu pour être NP-complet [Kar72]. La preuve est faite en trois étapes : nous montrons tout d'abord que PDFD est dans NP, puis que SCP peut être réduit en PDFD, et finalement que cette réduction peut être faite en temps polynomial.

Tout d'abord, montrons que PDFD est dans NP. Pour vérifier qu'une solution f_d à PDFD est valide, il suffit de vérifier que la fonction f_d satisfait la définition des protocoles de routage ρ -compatibles. Cette vérification nécessite $\mathcal{O}(|V|)$, ce qui est bien polynomial.

À présent, réduisons SCP à PDFD. Rappelons que SCP considère un ensemble de valeurs U et une famille $S = \{s_i\}_i$ telle que pour tout i , $s_i \subset U$, et consiste à trouver un sous-ensemble de S tel que chaque élément de U est couvert au moins une fois. Considérons une instance quelconque de SCP. La réduction que nous proposons consiste à créer un graphe orienté où chaque $s_i \in S$ est associé à un ensemble de nœuds, et où chaque élément de U correspond à une boucle.

- *Construction du graphe.* Construisons le graphe orienté $G = (V, E)$ de la manière suivante (voir la figure 3.22 pour un exemple). On considère dans E deux types de liens : les liens de type \mathcal{R}_1 (représentés en traits pleins sur la figure) et les liens de type \mathcal{R}_2 (représentés en traits pointillés). À chaque famille $s_i \in S$, on associe plusieurs nœuds dans V , notés n_i^j , pour j allant de 1 à $|s_i| + 2$ ($|s_i|$ représentant le cardinal de s_i). n_i^1 est relié à n_i^2 par un lien de type \mathcal{R}_1 , et n_i^j est relié à n_i^{j+1} par un lien de type \mathcal{R}_2 , pour tout $j \in [2; |s_i| + 1]$. Pour chaque $u \in U$, une boucle est construite entre toutes les familles de S qui contiennent ce u (pour simplifier la présentation, les boucles relient les familles de S dans l'ordre croissant des indices). Pour tout i et pour tout $j \in [1; |s_i|]$, n_i^{j+2} est connecté à n_1^k par un lien de type \mathcal{R}_1 , tel que si u est le j -ème élément de s_i , k est soit le plus petit indice supérieur à i tel que $u \in s_k$ (s'il existe), soit le plus petit indice tel que $u \in s_k$. Finalement, G possède un sommet supplémentaire, d . Pour chaque nœud n_i^j qui ne possède pas de lien de type \mathcal{R}_1 (respectivement \mathcal{R}_2), on ajoute un lien de type \mathcal{R}_1 (respectivement \mathcal{R}_2) entre ce nœud n_i^j et d . La figure 3.22 présente le graphe obtenu pour une instance particulière de SCP. La figure 3.23 reprend le même exemple, en mettant en avant la boucle correspondant à $u = 3$. Comme u apparaît dans s_1 , s_3 et s_4 , un nœud n_1^a est relié à n_3^1 , un nœud n_3^b est relié à n_4^1 , et un nœud n_4^c est relié à n_1^1 . On remarque qu'il ne peut pas exister de boucle empruntant uniquement des liens de type \mathcal{R}_1 ou de type \mathcal{R}_2 . Pour s'en convaincre, il suffit de voir que les arêtes de type \mathcal{R}_1 connectent uniquement les nœuds n_i^1 aux nœuds n_i^2 , les nœuds n_i^k aux nœuds n_j^1 , avec $k \geq 3$ et $i \neq j$, et certains nœuds n_i^j à d . De plus, les arêtes de type \mathcal{R}_2 connectent uniquement les nœuds n_i^k aux nœuds n_i^{k+1} , avec $k \in [2; |s_i| + 1]$, et certains nœuds n_i^j à d .
- *Réduction.* Nous devons à présent montrer que si SCP a une solution pour un k donné, alors PDFD a une solution pour $\rho = (|S| - k)/|S|$, et que si SCP n'a pas de solution pour un k donné, alors PDFD n'a pas de solution pour $\rho = (|S| - k)/|S|$.
- Supposons que l'instance de SCP a une solution $S' = \{s'_1, \dots, s'_k\} \subset S$. Tous

les nœuds de U sont couverts par S' , ce qui signifie que toutes les boucles du graphe (chacune correspondant à un élément $u \in U$) peuvent être couvertes par un ensemble de k nœuds (le premier nœud de chaque ensemble de nœuds correspondant aux $\{s'_i\}$). Il est possible de construire une fonction f_d qui ne produit aucun conflit, à l'exception de ces k nœuds. En effet, considérons le graphe orienté obtenu en enlevant chacun de ces k nœuds. Ce graphe est acyclique, car un nœud a été supprimé pour chacune des boucles du graphe. En construisant f_d au moyen d'un parcours topologique de ce graphe orienté acyclique, on obtient que $f_d(n) > f_d(\mathcal{R}_1(n, d))$ et que $f_d(n) > f_d(\mathcal{R}_2(n, d))$, ce qui est implique que $f_d(n) > \max\{f_d(\mathcal{R}_1(n, d)), f_d(\mathcal{R}_2(n, d))\}$, pour tous les nœuds sauf les k supprimés. \mathcal{R}_1 et \mathcal{R}_2 sont bien ρ -compatibles avec $\rho = (|S| - k)/|S|$. Sur l'exemple de la figure 3.22, la famille $S' = \{s_1, s_2\}$ permet de couvrir U avec $k = 2$. Le graphe obtenu en supprimant les nœuds n_1^1 et n_2^1 est représenté sur la figure 3.24 : on peut vérifier que ce graphe est bien acyclique.

- Supposons que l'instance de SCP n'a pas de solution pour un k donné. Il n'est donc pas possible de couvrir tous les éléments de U avec k ensembles de S , ce qui signifie qu'il n'est pas possible de couvrir toutes les boucles du graphe avec k nœuds. Même si ces k nœuds sont en conflit, il reste des boucles du graphe qui ne sont pas couvertes : on ne peut pas trouver une fonction f_d valide sur une topologie qui contient des boucles. Ainsi, on ne peut pas trouver de protocoles \mathcal{R}_1 et \mathcal{R}_2 qui soient 1-compatibles en enlevant seulement k nœuds, donc les protocoles ne sont pas ρ -compatibles avec $\rho \geq (|S| - k)/|S|$, et PDFD n'a pas de solution.

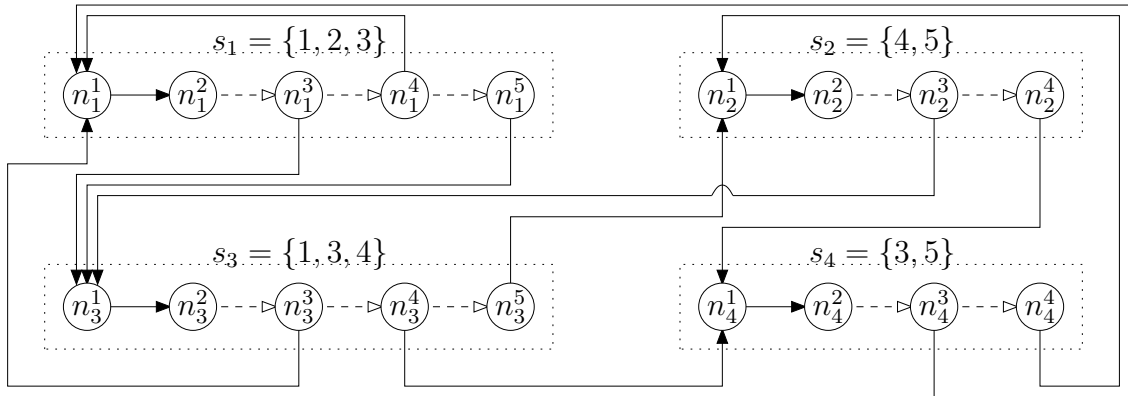


Figure 3.22 – Transformation de l'instance $S = \{\{1, 2, 3\}, \{4, 5\}, \{1, 3, 4\}, \{3, 5\}\}$ de SCP en une instance de PDFD. Le nœud d n'est pas représenté (par souci de clarté).

Finalement, montrons que la réduction de SCP à PDFD peut être faite en temps polynomial. Chaque famille s_i de S est transformée en $|s_i| + 2$ nœuds dans G , ce qui nécessite $\mathcal{O}(|S| \cdot |U|)$ opérations. Pour chaque élément u de U , on construit une boucle entre des nœuds de G , chaque boucle ayant autant de liens qu'il y a de familles de S contenant u . Chaque boucle nécessite donc $\mathcal{O}(|S|)$ opérations, ce qui aboutit à un total de $\mathcal{O}(|S| \cdot |U|)$ opérations pour construire toutes les boucles. La complexité totale de la transformation est donc $\mathcal{O}(|S| \cdot |U|)$, ce qui est bien polynomial. \square

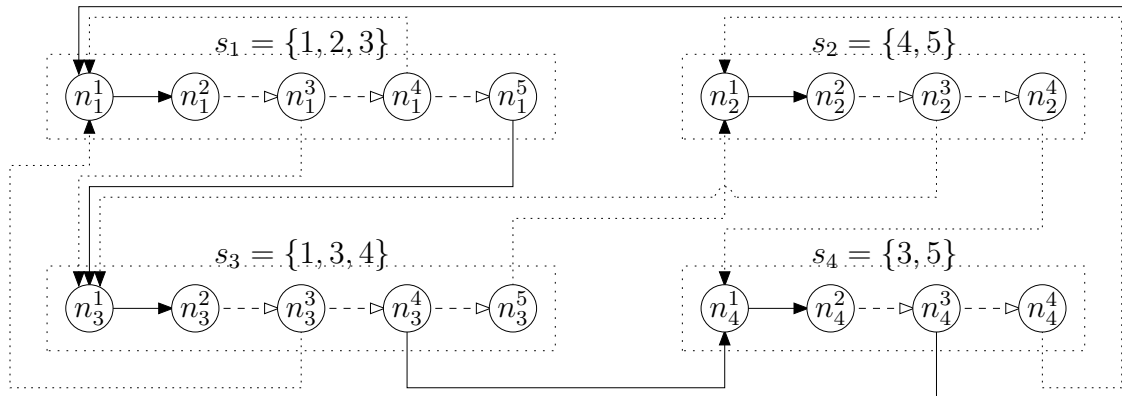


Figure 3.23 – Reprise de l'exemple de la figure 3.22, dans lequel la boucle $u = 3$ est mise en avant. Cette boucle relie un nœud correspondant à chaque ensemble $s_1 = \{1, 2, 3\}$, $s_3 = \{1, 3, 4\}$ et $s_4 = \{3, 5\}$.

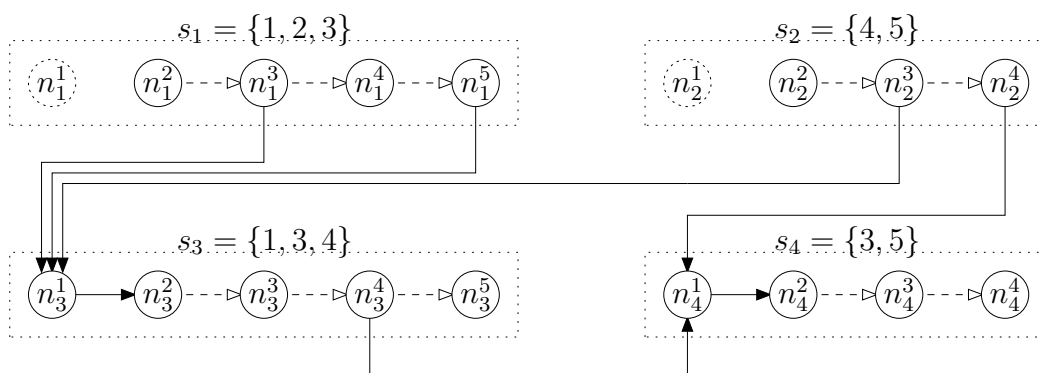


Figure 3.24 – Reprise de l'exemple de la figure 3.22, dans lequel les nœuds correspondant à la solution $S' = \{s_1, s_2\}$ ont été supprimés. Le graphe résultant est bien acyclique.

L'heuristique de modification de la topologie (HMT) [ERGM13b*] est une heuristique visant à réduire le nombre de nœuds en conflit (c'est-à-dire, visant à augmenter le pourcentage ρ pour deux protocoles ρ -compatibles).

HMT fonctionne de la manière suivante. À partir de la connaissance de la topologie $G = (V, E)$, et de la fonction de distance $f_d^G(n)$ donnant le nombre de sauts sur G de n à d selon le protocole de routage \mathcal{R}_1 (ou \mathcal{R}_2 , arbitrairement³⁰). HMT calcule le nombre de nœuds en conflit pour le graphe G selon f_d^G . Puis, pour chaque arête $e \in E$, HMT construit le graphe $G_e = (V, E \setminus \{e\})$ et, lorsque G_e est connexe, calcule le nombre de nœuds en conflit pour G_e selon $f_d^{G_e}$. Finalement, HMT choisit le graphe G' (avec $G' = G$ ou $G' = G_{e'}$ pour une certaine arête $e' \in E$) qui produit le nombre minimum de nœuds en conflit, et informe les protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 de ne pas utiliser l'éventuelle arête e' . La fonction de décision est la fonction $f_d^{G'}$.

Quand HMT est utilisée, une arête (au plus) est supprimée de la topologie de manière logique. Même si cette suppression peut légèrement augmenter la distance des routes pour les protocoles de routage, l'objectif de HMT est de réduire significativement le nombre de nœuds en conflit dans la nouvelle topologie. Les inconvénients principaux de HMT sont qu'elle nécessite de connaître la topologie complète, requiert de pouvoir calculer les prochains sauts selon \mathcal{R}_1 ou \mathcal{R}_2 sur une nouvelle topologie, et requiert une grande capacité de calcul pour le nœud qui calcule G' .

3.3.2 Protocoles retardables

Comme certains protocoles de routage ne sont pas compatibles³¹ et que des boucles peuvent apparaître pour les protocoles ρ -compatibles (avec $\rho < 1$), il est important de disposer d'une propriété moins stricte.

Définition 11 (Protocoles de routage retardables [ERGM11*]). Deux protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 sont dits retardables quand certains nœuds peuvent conserver un paquet un temps fini pour éviter les boucles.

Deux protocoles de routage quelconques peuvent toujours être considérés retardables. En effet, si les deux protocoles sont \mathcal{R}_1 et \mathcal{R}_2 , si chaque nœud considère les deux protocoles alternativement après un temps fini, et si tous les nœuds prennent la décision d'ignorer \mathcal{R}_2 (ou respectivement, \mathcal{R}_1), les paquets seront acheminés selon \mathcal{R}_1 uniquement (respectivement \mathcal{R}_2), ce qui ne causera pas de boucles. Cette solution triviale est toutefois à exclure en pratique, puisqu'elle revient à supprimer complètement l'un des deux protocoles de routage, ce qui est contraire au principe d'en disposer de deux.

Dans la suite, nous allons étudier des règles qui permettent aux nœuds d'acheminer selon \mathcal{R}_1 ou \mathcal{R}_2 arbitrairement, à quelques nœuds près.

3.3.2.1 Caractérisation des protocoles retardables

Théorème 3. *Soit un graphe orienté $G = (V, E)$, $d \in V$ une destination, et \mathcal{R}_1 et \mathcal{R}_2 deux protocoles de routage qui alternent en temps fini. S'il existe une fonction*

³⁰. HMT est plus efficace quand le protocole produisant les plus longs chemins est utilisé pour f_d^G .

³¹. C'est d'ailleurs le cas pour la plupart des protocoles.

de décision f_d telle que $f_d(d) = 0$ et $\forall n \in V \setminus \{d\}$, on a $f_d(n) > \min\{f_d(\mathcal{R}_1(n, d)), f_d(\mathcal{R}_2(n, d))\}$, alors \mathcal{R}_1 et \mathcal{R}_2 sont retardables en utilisant f_d . Lorsque la fonction de décision f_d détermine que le paquet ne peut pas être transmis par un protocole de routage, le nœud conserve le paquet jusqu'à ce que le protocole de routage change [ERGM11*, ERGM13a*].

Démonstration. Cette preuve étant très similaire à celle du théorème 1, seules les parties différentes sont reprises. La différence principale avec le théorème 1 est qu'il faut à présent prendre en compte le fait qu'un nœud puisse conserver un paquet.

La construction du chemin p à partir de la séquence de décisions $(r_i)_i$ se fait ainsi. $p = (n_0, n_1, n_2, \dots)$, avec $n_0 = n$ et pour tout nœud $n \in V \setminus \{d\}$:

- si $f_d(n_i) \leq f_d(\mathcal{R}_{r_i}(n_i, d))$, alors n_i conserve le paquet, et $n_{i+1} = n_i$ (ce qui est un retard, et non pas une boucle),
- sinon, $n_{i+1} = \mathcal{R}_{r_i}(n_i, d)$.

On peut montrer que si p est infini, tous les nœuds de p appartiennent à $V \setminus \{d\}$. Montrons que p est fini par contradiction, en supposant que p est infini. Construisons la séquence $s = (f_d(n_0), f_d(n_1), f_d(n_2), \dots)$ en appliquant f_d à chaque nœud de p . Pour chaque nœud n_i de p , comme $n_i \in V \setminus \{d\}$, nous avons $f_d(n_i) \geq f_d(n_{i+1})$. En effet :

- Si $f_d(n_i) \leq f_d(\mathcal{R}_{r_i}(n_i, d))$, le paquet est conservé, et donc $n_{i+1} = n_i$. Dans ce cas, $f_d(n_{i+1}) = f_d(n_i)$.
- Sinon, $f_d(n_i) > f_d(\mathcal{R}_{r_i}(n_i, d)) = f_d(n_{i+1})$.

La séquence s est décroissante et constante par morceaux. Nous allons montrer que s ne conserve la même valeur que lorsqu'un nœud conserve le paquet, ce qui signifie que s ne conserve pas la même valeur pendant un temps infini (puisque les protocoles alternent en temps fini). Pour cela, nous allons montrer que si un nœud n_i décide de conserver le paquet, il existe $j > i$ tel que $f_d(n_j) > f_d(\mathcal{R}_{r_j}(n_j, d))$. Comme les protocoles alternent en temps fini, $(r_i)_i$ ne contient pas une suite infinie de valeurs consécutives identiques. Soit j le plus petit entier supérieur à i tel que $r_j \neq r_i$. Comme j est le plus petit entier valide, nous avons $n_i = n_{i+1} = \dots = n_j$. Comme n_i a décidé de conserver le paquet, nous avons $f_d(n_i) \leq f_d(\mathcal{R}_{r_i}(n_i, d))$. Comme $n_i = n_j$, nous avons $f_d(n_j) \leq f_d(\mathcal{R}_{r_i}(n_j, d))$. Comme $r_j \neq r_i$, on a $\{r_i, r_j\} = \{1, 2\}$. Par définition de f_d , nous avons $f_d(n_j) > \min\{f_d(\mathcal{R}_1(n_j, d)), f_d(\mathcal{R}_2(n_j, d))\}$, ce qui peut se réécrire en $f_d(n_j) > \min\{f_d(\mathcal{R}_{r_i}(n_j, d)), f_d(\mathcal{R}_{r_j}(n_j, d))\}$ (en remarquant que \mathcal{R}_1 et \mathcal{R}_2 ont été remplacés par \mathcal{R}_{r_i} et \mathcal{R}_{r_j}). Nous avons donc $f_d(n_j) > f_d(\mathcal{R}_{r_i}(n_j, d))$ ou $f_d(n_j) > f_d(\mathcal{R}_{r_j}(n_j, d))$. Or, nous avons montré plus tôt que $f_d(n_j) \leq f_d(\mathcal{R}_{r_i}(n_j, d))$, donc nous avons bien $f_d(n_j) > f_d(\mathcal{R}_{r_j}(n_j, d))$. Il est impossible d'avoir une séquence s d'entiers positifs, infinie et décroissante, mais qui ne reste jamais constante pendant un temps infini : le chemin p est donc fini.

On peut montrer que p atteint d . Montrons à présent que p ne contient pas de boucle.

- Montrons que s'il existe x tel que $n_x = n_{x+1}$ dans p , c'est que le paquet a été conservé par n_x . Par contradiction, supposons que $n_x = n_{x+1}$, mais que le paquet n'a pas été conservé par n_x . Nous avons alors $f_d(n_x) > f_d(\mathcal{R}_{r_x}(n_x, d)) = f_d(n_{x+1})$. Or $n_x = n_{x+1}$, donc $f_d(n_x) = f_d(n_{x+1})$, ce qui est contradictoire.
- Montrons maintenant qu'un même nœud n'achemine pas le même paquet deux fois (conservation exclue). Par contradiction, faisons l'hypothèse qu'il existe

trois indices $x < y < z$ tels que $n_x = n_z$ (n_x est donc le nœud ayant acheminé le même paquet deux fois) et $n_x \neq n_y$ (c'est-à-dire que n_x n'a pas conservé le paquet). Construisons la séquence $s = (f_d(n_0), f_d(n_1), f_d(n_2), \dots, f_d(n_x), \dots, f_d(n_y), \dots, f_d(n_z))$. On peut considérer deux cas.

- Si $n_z = d$, alors $n_x = d$. Dans ce cas, le prochain saut de n_x n'est pas défini, et p ne peut pas passer par n_y , ce qui est une contradiction.
- Si $n_z \neq d$, alors p ne termine pas en n_z . Dans ce cas, chaque nœud de p de n_0 à n_z appartient à $V \setminus \{d\}$. Ainsi, pour tout nœud $i \in [0; z]$, nous avons $f_d(n_i) \geq f_d(\mathcal{R}_{r_i}(n_i, d)) \geq f_d(n_{i+1})$. De plus, comme n_x n'a pas conservé le paquet, nous avons $f_d(n_x) > f_d(n_{x+1})$, ce qui signifie que $f_d(n_x) > f_d(n_y)$ (car $f_d(n_y) \geq f_d(n_{x+1})$). De manière similaire, comme $n_y \neq n_z$ (car $n_y \neq n_x$ et $n_x = n_z$), nous avons $f_d(n_y) > f_d(n_z)$. Nous avons finalement $f_d(n_x) > f_d(n_z)$, ce qui est contradictoire car nous avons supposé que $n_z = n_x$.

Dans les deux cas, nous obtenons une contradiction. L'hypothèse que p passe deux fois par le même nœud n'est pas valide. □

Propriété 9. *Soit un graphe orienté $G = (V, E)$, $d \in V$ une destination, deux protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 qui alternent en temps fini, et f_d^1 la distance à d selon \mathcal{R}_1 et f_d^2 la distance à d selon \mathcal{R}_2 . \mathcal{R}_1 et \mathcal{R}_2 sont retardables en utilisant f_d^1 (respectivement, f_d^2).*

Démonstration. Soit un graphe orienté $G = (V, E)$, $d \in V$ une destination, deux protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 qui alternent en temps fini, f_d^1 la distance à d selon \mathcal{R}_1 . Comme f_d^1 est une distance, $f_d^1(d) = 0$. Pour tout $n \in V \setminus \{d\}$, on a $f_d^1(n) > f_d^1(\mathcal{R}_1(n, d)) \geq \min\{f_d^1(\mathcal{R}_1(n, d)), f_d^1(\mathcal{R}_2(n, d))\}$. Ainsi, \mathcal{R}_1 et \mathcal{R}_2 sont aussi retardables en utilisant f_d^1 . □

Par exemple, le protocole de routage hiérarchique et les protocoles des plus courts chemins sont retardables en utilisant pour fonction de décision la distance sur l'arbre (ou la plus courte distance). Le protocole de routage raccourci et les protocoles des plus courts chemins sont retardables en utilisant pour fonction de décision la distance sur l'arbre³² (ou la plus courte distance).

3.3.2.2 Utilisation de protocoles retardables

Jusqu'à présent, les propriétés que nous avons obtenues sur les protocoles compatibles et retardables ont deux inconvénients principaux. D'une part, il peut être difficile de trouver une fonction de décision qui rend les protocoles compatibles ou retardables (voire, de la calculer au niveau MAC). D'autre part, certains nœuds peuvent conserver longtemps les paquets (avant que le protocole de routage ne change). Dans la suite, nous allons chercher à résoudre ces inconvénients.

32. La distance sur l'arbre n'est pas la fonction de distance utilisée par le protocole de routage raccourci : l'exemple donné n'est donc pas une conséquence directe de la propriété 9. Toutefois, on peut remarquer que puisque la distance utilisée par le protocole de routage raccourci est toujours inférieure ou égale à la distance sur l'arbre, la propriété reste correcte.

Modération de la fonction de décision.

L'heuristique de protocoles retardables probabiliste (PRP) [ERGM13b*] est une heuristique probabiliste qui permet de modérer les contraintes imposées par la fonction de décision f_d , c'est-à-dire, dans certains cas, de router les paquets au prochain saut même si la fonction de décision l'interdit.

PRP fonctionne de la manière suivante. Quand un nœud détermine qu'il doit attendre pour transmettre un paquet pour éviter de l'envoyer dans une boucle, PRP peut néanmoins autoriser le nœud à transmettre le paquet. Cette autorisation est donnée de manière aléatoire, selon une probabilité fixée à l'avance. L'heuristique PRP se base sur l'intuition que les boucles évitées par la fonction de décision sont en pratique peu probables, et que les boucles sont causées par des choix répétitifs de décisions de routage (qui peuvent être rendus moins probable par des choix aléatoires de décisions de routage). Il est possible que PRP crée des boucles de routage (mais, statistiquement non infinies à cause de la nature probabiliste de PRP), mais il est aussi possible que le temps d'acheminement de certains paquets soit réduit (puisque les nœuds conservent en moyenne les paquets moins longtemps avec PRP) [ERGM13b*].

Superposition de protocoles de routage.

La méthode de superposition de protocoles de routage consiste à profiter d'un protocole de routage particulier \mathcal{R}^* , dont la fonction de distance est connue, pour rendre des protocoles retardables (même si leur fonction de distance est inconnue ou dure à calculer).

Définition 12 (Superposition de deux protocoles de routage). Soit $G = (V, E)$ un ensemble de nœuds, $d \in V$ une destination, \mathcal{R} et \mathcal{R}^* deux protocoles de routage quelconques. La superposition de \mathcal{R} et de \mathcal{R}^* , notée $\mathcal{R} + \mathcal{R}^*$, fonctionne de la manière suivante. Soit f_d^* la fonction de distance de \mathcal{R}^* . Pour router un paquet à la destination d , tout nœud n détermine si $f_d^*(n) > f_d^*(\mathcal{R}(n, d))$. Si c'est le cas, le paquet est envoyé selon \mathcal{R} . Sinon, le paquet est envoyé selon \mathcal{R}^* .

Propriété 10. Soit un graphe orienté $G = (V, E)$, $d \in V$ une destination, et trois protocoles de routage : \mathcal{R}_1 , \mathcal{R}_2 et \mathcal{R}^* . Soit f_d^* une fonction de décision telle que $f_d^*(d) = 0$ et $\forall n \in V \setminus \{d\}$, $f_d^*(\mathcal{R}^*(n, d)) < f_d^*(n)$. Alors, \mathcal{R}_1 et $\mathcal{R}_2 + \mathcal{R}^*$ sont retardables en utilisant la fonction f_d^* [ERGM11*].

Démonstration. Selon le théorème 3, il suffit de montrer que $\forall n \in V \setminus \{d\}$, on a $f_d^*(n) > \min\{f_d^*(\mathcal{R}_1(n, d)), f_d^*((\mathcal{R}_2 + \mathcal{R}^*)(n, d))\}$. Nous allons montrer que $f_d^*(n) > f_d^*((\mathcal{R}_2 + \mathcal{R}^*)(n, d))$.

- Si $f_d^*(n) > f_d^*(\mathcal{R}_2(n, d))$, le nœud n utilise le protocole \mathcal{R}_2 pour acheminer le paquet. Ainsi, $f_d^*((\mathcal{R}_2 + \mathcal{R}^*)(n, d)) = f_d^*(\mathcal{R}_2(n, d))$, et on en déduit que $f_d^*(n) > f_d^*((\mathcal{R}_2 + \mathcal{R}^*)(n, d))$.
- Si $f_d^*(n) \leq f_d^*(\mathcal{R}_2(n, d))$, le nœud n utilise le protocole \mathcal{R}^* pour acheminer le paquet. Ainsi, $f_d^*((\mathcal{R}_2 + \mathcal{R}^*)(n, d)) = f_d^*(\mathcal{R}^*(n, d))$. D'après la définition de f_d^* , on a $f_d^*(n) > f_d^*(\mathcal{R}^*(n, d))$, et donc $f_d^*(n) > f_d^*((\mathcal{R}_2 + \mathcal{R}^*)(n, d))$.

Les deux protocoles \mathcal{R}_1 et $\mathcal{R}_2 + \mathcal{R}^*$ sont donc retardables. \square

En pratique, le choix de \mathcal{R}^* se fait selon plusieurs critères. La fonction f_d^* doit être facile à calculer, pour tout d . De plus, la surcharge de \mathcal{R}^* par rapport au protocole \mathcal{R}_2 , en termes de messages de contrôle ou d'énergie par exemple, doit être limitée³³. Nous conseillons d'utiliser pour protocole \mathcal{R}^* un protocole utilisant explicitement f_d^* pour calculer les routes (comme c'est le cas pour le protocole de routage hiérarchique, ou pour les protocoles de routage des plus courts chemins).

3.4 Conclusions

Dans un même réseau de capteurs sans fil, plusieurs applications cohabitent généralement : certaines applications collectent des données périodiquement, tandis que d'autres réagissent à des commandes venant d'opérateurs distants, ou surveillent des événements significatifs. Chaque application produit différents types de trafic, chacun ayant des propriétés spécifiques, et il est souvent difficile de fournir à chaque trafic une qualité de service adaptée lorsque l'on utilise une seule pile protocolaire. Les architectures multi-piles, comme la pile Ocari présentée dans la partie 3.1, répondent à ce problème en permettant l'utilisation de plusieurs piles protocolaires dans un même réseau.

Dans les architectures multi-piles simples, étudiées dans la partie 3.2.1, l'application du nœud source décide de la pile protocolaire la plus appropriée pour chaque paquet, et le paquet est pris en charge par cette même pile jusqu'à la destination. Il n'y a donc pas d'interaction entre les piles protocolaires. Ces architectures multi-piles simples permettent de fournir un premier niveau de différenciation de services, mais souffrent d'un manque de flexibilité (puisque les piles protocolaires sont indépendantes) et parfois d'un problème de délai (lorsque les piles protocolaires doivent être activées à tour de rôle).

Dans les architectures multi-piles locales, étudiées dans la partie 3.2.2, chaque nœud décide de la pile la plus appropriée pour chaque paquet, en fonction d'un contexte local au nœud. Les interactions entre piles protocolaires sont importantes. Ces architectures multi-piles locales permettent de tirer profit des avantages de chacune des piles protocolaires, mais introduisent le risque de boucles de routage.

Les boucles de routage apparaissent quand les protocoles de routage utilisés ne vérifient pas certaines propriétés de compatibilité, que nous avons proposées lors de la thèse de Nancy EL RACHKIDY, et qui sont détaillées dans la partie 3.3. Nous avons proposé des solutions locales à chaque nœud qui permettent d'éviter ces boucles, quitte à ce que certains paquets soient retardés. Une étude de l'impact de ces boucles sur la longueur des chemins est faite dans [ERGM13b*].

L'originalité des propriétés de compatibilité est qu'il s'agit de propriétés d'ensembles de protocoles, alors qu'il est plus fréquent de voir dans la littérature les propriétés des protocoles considérés indépendamment. Le raisonnement sur des ensembles de protocoles (plutôt que sur des protocoles indépendants) nous semble être

33. Il est d'ailleurs possible que la surcharge de \mathcal{R}^* soit presque négligeable. Par exemple, si \mathcal{R}_2 est un protocole basé sur une topologie en arbre, un routage hiérarchique peut être utilisé pour \mathcal{R}^* : ainsi, la surcharge de la maintenance de l'arbre est partagée par \mathcal{R}_2 et par \mathcal{R}^* , et la fonction f_d^* est simple à calculer.

une piste de recherche très intéressante.

D'autres perspectives concernent aussi les architectures multi-piles locales. Il serait intéressant d'étudier les architectures dans lesquelles la réunion des piles n'a ni lieu au niveau de la couche liaison de données, ni au niveau de la couche réseau. De même, il serait judicieux d'étudier le gain des architectures multi-piles locales par rapport aux architectures multi-piles simples où le calendrier est dimensionné dynamiquement en fonction du trafic. Finalement, nous pourrions aussi étudier la possibilité de faire des échanges de paquets entre piles protocolaires en fonction de l'énergie résiduelle des nœuds (ou d'autres critères).

CHAPITRE

4

DÉCISION GLOBALE DE CHANGEMENT DE PILES PROTOCOLAIRES

Pour répondre aux besoins diversifiés des applications, nous avons envisagé dans le chapitre 3 l'utilisation d'architectures utilisant plusieurs piles protocolaires. Le passage d'une pile protocolaire à l'autre se faisait sur des critères locaux aux nœuds : soit une seule pile était utilisée pour un paquet donné, sur l'ensemble du chemin (c'était le cas des architectures simples), soit chaque nœud choisissait la pile à utiliser indépendamment des autres nœuds (c'était le cas des architectures locales).

Toutefois, le choix de la pile protocolaire à utiliser ne peut pas toujours être fait pour chaque paquet ou par chaque nœud de manière indépendante. Dans certains cas, la pile à utiliser est déterminée par des nœuds distants, suite à des décisions applicatives généralement. Par exemple, un opérateur peut décider de basculer les nœuds sur une nouvelle version d'un protocole de routage (probablement plus efficace), ou les nœuds peuvent basculer dans un mode de suivi plus adapté aux évènements détectés. La transition d'une pile protocolaire à une autre ne se fait pas de manière instantanée, mais prend un certain temps à cause du délai d'acheminement de la notification de changement. Il est important de proposer des mécanismes permettant à ce changement de pile de se réaliser sans dégradation significative des performances du réseau durant la période transitoire.

Dans la partie 4.1, nous introduisons le projet Clervolc, dont l'objectif est de concevoir un ensemble de protocoles pour un réseau de capteurs sans fil surveillant des phénomènes géophysiques, et notamment les séismes à proximité des volcans. Le réseau est supposé être opérationnel pendant plusieurs mois (voire plusieurs années). Pour économiser l'énergie des nœuds, deux niveaux de surveillance sont envisagés :

un niveau de surveillance périodique (avec des mesures peu fréquentes) et un niveau de surveillance intensive (avec des mesures très fréquentes, en cas de détection d'un séisme par exemple). Chaque niveau de surveillance nécessite l'utilisation de protocoles adaptés. Les changements de niveaux sont initiés par les nœuds détectant des événements significatifs ou par le puits applicatif (relié à une station de contrôle, de laquelle peuvent partir des commandes d'une instance de décision ou d'un opérateur). Dans la partie 4.2, nous généralisons cette approche en proposant les architectures multi-piles globales, dans lesquelles un paquet peut être acheminé selon plusieurs piles protocolaires, en fonction d'une décision globale (c'est-à-dire pouvant être externe à la source du paquet ou au nœud qui achemine le paquet). Nous montrons que les problématiques des architectures multi-piles globales apparaissent pendant les périodes de transition d'un niveau à un autre et nous montrons comment éviter les boucles de routage qui peuvent survenir. Dans la partie 4.3, nous revenons sur des aspects fondamentaux des architectures multi-piles globales, et nous proposons une étude de la manière dont les nœuds changent d'état en fonction des changements de niveaux de surveillance. Nous proposons notamment un algorithme permettant d'évaluer le temps de transition nécessaire pour passer d'un niveau à un autre.

4.1 Pré-étude d'un cas concret : le projet Clervolc

Le projet Clervolc (Centre Clermontois de Recherche sur le Volcanisme) [Cle13] a pour but d'innover dans le contexte de la surveillance volcanologique. Le projet est financé par un Labex (depuis 2011), et regroupe six laboratoires de recherche (le LAMP, le LAPSCO, le LIMOS, le LM, le LMV, et le LPC) et deux services de recherche et développement (le BRGM et l'OPGC). Il s'agit d'un projet prévu pour durer une dizaine d'années, et qui n'en est qu'à son démarrage.

Le projet Clervolc est découpé en huit programmes : la surveillance opérationnelle de l'activité volcanique et des panaches associés, la structure interne et la déformation des édifices volcaniques, les éléments volatils : la force motrice de l'activité volcanique, l'évolution des édifices volcaniques : mécanismes et échelles de temps, le contrôle mantellique profond des sources des volcans, les écoulements d'origine volcanique : genèse et impact, une approche plus quantitative des risques volcaniques et de leur impact sociétal, et les réseaux de capteurs sans fil pour la surveillance des volcans. Ces huit programmes visent à concevoir des outils permettant de mieux comprendre et de mieux surveiller les volcans.

4.1.1 Objectifs de la surveillance par réseau de capteurs sans fil

La surveillance de volcans par des capteurs est réalisée depuis de nombreuses années, en utilisant généralement des sismographes [McN96] ou des images satellites [TBZ⁺05]. Les mesures produites par ces capteurs fournissent des informations importantes sur les déformations de la structure du volcan, la position de l'épicentre des séismes, leur intensité, et permettent ainsi de suivre l'évolution des volcans.

Les réseaux de capteurs sans fil constituent une technologie prometteuse pour les nouvelles applications de surveillance. En effet, ces réseaux permettent d'obtenir des données en temps réel provenant d'endroits considérés auparavant comme inaccessibles (sur la trajectoire d'un écoulement de lave ou d'un éboulement, à l'intérieur d'un panache opaque, au sommet du volcan, etc.). Cela est dû au faible coût des nœuds capteurs (qui vont même jusqu'à être considérés comme des consommables¹) et à leur capacité de communication sans fil (qui leur permet de communiquer même lorsqu'ils sont déplacés ou recouverts, au détriment d'une dégradation des performances).

4.1.1.1 Approches existantes

La surveillance de volcans par des réseaux de capteurs sans fil, c'est-à-dire par des capteurs autonomes en énergie et reliés entre eux par des liaisons sans fil (à faible portée), est très récente. Dans la suite, nous décrivons les principales approches de surveillance de volcans de la littérature.

Retour d'expérience sur le Tungurahua (Équateur).

Dans [WAJR⁺05], les auteurs décrivent le premier réseau de capteurs sans fil déployé sur un volcan : le Tungurahua, en Équateur. Les auteurs décrivent le déploiement du réseau, montrent qu'il a été opérationnel pendant 54 heures, et proposent un algorithme permettant de faire de la détection distribuée d'évènements afin de réduire le trafic en n'envoyant que les données corrélées.

Ce premier réseau est constitué de trois nœuds capteurs (munis d'antennes d'environ 17 cm), tous les trois disposés à moins de 11 m d'un nœud agrégateur (muni d'une antenne d'un mètre environ, avec un gain de 9 dBi). Tous ces nœuds sont intégrés dans des boîtiers étanches. Les capteurs mesurent des ondes acoustiques basses fréquences, et les nœuds capteurs transmettent des paquets contenant 25 mesures consécutives (pour une taille de 32 octets) au nœud agrégateur, 4 fois par seconde. Les communications entre les nœuds capteurs et le nœud agrégateur se font dans la bande de fréquences 433 MHz, et la communication entre le nœud agrégateur et le centre de surveillance (distant de neuf kilomètres) se fait dans une bande de fréquences proche de 900 MHz.

Le protocole MAC utilisé pour les communications entre les nœuds capteurs et le nœud agrégateur est le protocole par défaut de TinyOS² (qui est un protocole hybride entre BMAC [PHC04] et XMAC [BYAH06]). Le protocole de routage utilisé est basique, étant donné que la topologie est une étoile (non multi-saut). Les données issues des capteurs sont estampillées grâce à un GPS installé dans le nœud agrégateur, qui transmet périodiquement l'heure à tous les nœuds capteurs.

L'inconvénient majeur de la solution décrite dans [WAJR⁺05] est le faible nombre de nœuds capteurs, et la simplicité de la topologie.

1. En d'autres termes, les nœuds capteurs sont considérés comme moins importants que les données qu'ils produisent, et il devient donc acceptable de perdre un certain nombre de ces nœuds.

2. TinyOS [Uni13] est un système d'exploitation *open-source* nécessitant peu de ressources pour s'exécuter, et conçu pour les équipements sans fil à basse consommation.

Retour d'expérience sur l'Etna (Italie).

Dans [Zha05], l'auteur décrit une solution matérielle pour les nœuds d'un réseau de capteurs sans fil déployés sur l'Etna, en Italie.

Le réseau est constitué de trois nœuds capteurs (munis d'antennes ayant un gain de 12 dBi, mesurant 1,2 m par 41,9 cm, et placées à environ un mètre du sol), disposés à trois kilomètres les uns des autres, et tous à portée. L'un des nœuds capteurs est connecté par un câble série à une station de base. Les capteurs mesurent le spectre ultra-violet afin d'en déduire la densité en gaz des panaches volcaniques. Les communications se font dans la bande de fréquences 410 MHz. L'auteur indique que le protocole de routage utilisé se base sur des groupes de nœuds déterminés statiquement par la station de base, et que la station de base envoie aux nœuds leur table de routage complète³. Dans la topologie déployée, il est probable que le routage soit direct.

Les inconvénients majeurs de la solution décrite dans [Zha05] sont l'absence de résultats quantitatifs sur le déploiement, et la taille importante des antennes.

Retour d'expérience sur le Reventador (Équateur).

Dans [WALJ⁺06, WALW⁺06], les auteurs décrivent une expérience réalisée sur le Reventador, en Équateur, pendant 19 jours (avec deux changements de piles). Les auteurs proposent un protocole réactif de récupération des données en cas de détection d'un évènement sismique.

Le réseau est constitué de 16 nœuds capteurs (munis d'antennes similaires à celles de [Zha05], ayant un gain de 8 dBi et montées sur des mâts d'environ 1,5 m), disposés de manière relativement linéaire sur trois kilomètres, avec un espacement entre les nœuds de 200 m à 400 m. Les capteurs mesurent des ondes acoustiques et sismiques. Les communications se font sur la bande de fréquences 2,4 GHz. Un nœud passerelle communique avec un observatoire situé à quatre kilomètres en utilisant une antenne ayant un gain de 9 dBi.

Le protocole MAC utilisé est BMAC. Le protocole de routage utilisé est HopLQI [Zha05], qui est une adaptation du protocole MintRoute [WTC03]. HopLQI construit un arbre de routage en utilisant le LQI (*Link Quality Indicator*), qui est une mesure de la qualité d'un lien, et propage les messages par inondation. Pour la datation précise des évènements, un nœud équipé d'un GPS transmet l'heure aux autres nœuds en utilisant le protocole FTSP [MKSL04].

Les inconvénients de la solution décrite dans [WALJ⁺06, WALW⁺06] sont la faible durée de vie des nœuds capteurs, l'utilisation d'antennes volumineuses, et l'utilisation de la bande de fréquences 2,4 GHz, peu adaptée aux communications en extérieur et sans ligne de vue.

3. Le déploiement ne comportant que trois nœuds capteurs, il est probable que le réseau ne soit constitué que d'un seul groupe de nœuds, mais que le protocole réseau supporte davantage de groupes si certains nœuds se trouvent hors de portée de la station de base.

Retour d'expérience sur le Mont Saint Hélène (États-Unis).

Dans [SHX⁺09], les auteurs décrivent le déploiement d'un réseau de capteurs sans fil par hélicoptère sur le Mont Saint Hélène, aux États-Unis. Les auteurs évaluent la robustesse de leur réseau de capteurs sans fil sur une période d'un mois et demi.

Le réseau est composé de 5 nœuds capteurs. Chaque nœud capteur mesure environ 1,20 m de haut et pèse 35 kg. Les capteurs mesurent des ondes acoustiques et sismiques, ainsi que des changements de luminosité afin d'identifier les secousses liées aux impacts de foudre. Les communications se font sur la bande de fréquences 2,4 GHz, par l'intermédiaire d'antennes ayant un gain de 6 dBi.

Le protocole MAC utilisé est le protocole par défaut sur TinyOS. Le protocole de routage est MultihopOasis, proposé dans [SHX⁺09], qui s'appuie sur le protocole à vecteurs de distances MultihopLQI (implémenté de base dans TinyOS). La synchronisation est faite par le protocole Z-SYNC, proposé dans [SHX⁺09], qui combine l'utilisation d'un GPS par nœud capteur avec le protocole FTSP pour les aspects multi-sauts (lorsqu'un nœud n'a pas pu obtenir de signal GPS).

Les inconvénients de la solution décrite dans [SHX⁺09] sont l'utilisation d'un matériel coûteux pour les capteurs (rendant les nœuds capteurs non consommables), et l'utilisation de la bande de fréquences 2,4 GHz.

Retour d'expérience sur les simulations et tests en laboratoire.

Dans [CSL09], les auteurs présentent une solution qu'ils envisagent de déployer sur l'Ubinas, au Pérou. Les auteurs évaluent la qualité d'un microphone à infrasons, de leur mécanisme de synchronisation et de leur protocole de routage.

Le réseau envisagé est composé de 7 nœuds capteurs et d'un nœud passerelle. Les communications devraient avoir lieu sur la bande de fréquences 868 MHz avec une puissance d'émission de 8,6 dBm, avec l'objectif d'atteindre des portées allant de un kilomètre à cinq kilomètres en ligne de vue. Le nœud passerelle communiquerait sur la bande de fréquences 900 MHz avec l'observatoire situé à 65 km.

Le protocole MAC proposé utilise un accès en TDMA, avec un cycle d'activité de 3% (en l'absence de données). Chaque nœud capteur possède un GPS, qui est utilisé pour la synchronisation temporelle (en corrigeant la dérive). Le protocole de routage est D3 [DL05], qui construit un arbre et se base sur un gradient qui est fonction de la distance au puits. Les communications ont lieu pendant un intervalle de temps qui dépend de la distance au puits (afin de favoriser le transfert d'information des nœuds à la passerelle).

L'inconvénient principal de la solution décrite dans [CSL09] est l'absence de mesures sur le terrain qui permettraient de déterminer si les portées annoncées peuvent être atteintes en pratique⁴. De plus, la communication du nœud passerelle à l'observatoire est susceptible de consommer beaucoup d'énergie.

4. On peut d'ailleurs noter que la puissance de transmission de 8,6 dBm pour la bande de fréquences 868 MHz est au delà des conditions limites d'utilisation annoncées pour le composant radio utilisé (cf page 5 de [CC1020]). Le comportement de cette bande de fréquences dans ces conditions est donc à étudier avant un déploiement réel.

Retour d'expérience sur les algorithmes de détection.

Dans [Var09], l'auteur présente une amélioration de l'algorithme de détection d'évènements de [WALW⁺06] visant à réduire l'énergie consommée.

Dans [TXC⁺10], les auteurs conçoivent un algorithme collaboratif permettant aux nœuds du réseau de détecter les tremblements de terre (en obtenant leur nombre et leur durée) avec une bonne précision, tout en économisant l'énergie. Les résultats de simulation obtenus se basent sur les mesures fournies par [SHX⁺09].

Dans [LTZ⁺13], les auteurs conçoivent un algorithme collaboratif permettant aux nœuds de détecter les ondes sismiques primaires (qui sont essentielles pour comprendre les phénomènes qui se produisent dans les cheminées volcaniques). Les résultats de simulation obtenus se basent sur les mesures fournies par [SHX⁺09].

Les résultats présentés dans [Var09], [TXC⁺10] et [LTZ⁺13] ne concernent que des algorithmes de détection, et non pas des architectures de réseaux. Leur utilisation est donc complémentaire aux architectures et protocoles proposés dans la littérature (ou dans la suite de ce document).

4.1.1.2 Approche de l'équipe

L'équipe « réseaux et protocoles » du LIMOS intervient sur la conception d'un réseau de capteurs sans fil pour la surveillance volcanique. Cette thématique est partagée par plusieurs programmes du Labex Clervolc, et plus précisément, par les programmes de surveillance des panaches, de surveillance des déformations en surface, de surveillance des écoulements ou éboulements, et de mesures par un réseau de capteurs sans fil.

Les travaux présentés dans cette partie concernent les premières pistes étudiées dans le cadre du projet Clervolc. La plupart des travaux présentés sont en cours d'études, et il est possible que les architectures et protocoles présentés ne fassent pas tous partie des propositions retenues dans le cadre du projet Clervolc. Par exemple, il est possible que certains protocoles décrits dans la suite se basent sur des hypothèses qui sont trop contraignantes, ou que leur implémentation nécessite trop de ressources.

Notre application phare est la surveillance d'écoulements froids (comme les éboulements ou les lahars, qui sont des coulées de boue) ou chauds (comme les coulées de lave ou les écoulements pyroclastiques). Pour effectuer cette surveillance, nous considérons que les nœuds capteurs sont déployés sur la trajectoire de l'écoulement. Nous faisons l'hypothèse que les nœuds capteurs sont à bas prix, et peuvent être déplacés par l'action de l'écoulement ou détruits.

Nos objectifs sont multiples : nous envisageons de concevoir avec nos partenaires une plateforme matérielle adaptée au déploiement sur un volcan, de développer des protocoles adaptés (au niveau de la sous-couche MAC et de la couche réseau), de simuler ces protocoles, et de les valider en conditions réelles. Les protocoles que nous souhaitons réaliser doivent répondre à de nombreux critères. Ils doivent être économes en énergie afin de permettre un déploiement de plusieurs mois (voire, de plusieurs années). Ils doivent supporter un déploiement en conditions réelles, ce qui nécessite une adaptation aux conditions de propagation, un grand nombre de nœuds (pouvant aller jusqu'à quelques centaines), et la présence éventuelle de plusieurs

puits. Finalement, ils doivent pouvoir surveiller des évènements imprévisibles et discontinus, ce qui nécessite plusieurs niveaux de fonctionnement.

Nous envisageons de nous démarquer de l'état de l'art de la manière suivante.

- Les nœuds capteurs que nous utilisons seront à bas prix et de petite taille.
- Nous ferons le choix initial d'utiliser la bande de fréquences ISM à 433 MHz, adaptée aux communications à bas débit lorsque les nœuds sont partiellement enfouis ou en présence de végétation. Bien que les bandes de fréquences 2,4 GHz et 868 MHz soient plus classiques, elles nous semblent beaucoup moins robustes [ITU 833].
- Les protocoles réseaux que nous utiliserons seront adaptatifs (leurs caractéristiques et leurs performances pourront changer en fonction des évènements détectés) et dynamiques (ils seront conçus autour de l'hypothèse de la destruction possible de nœuds capteurs).
- Les données que nous collecterons ne se limiteront pas à la détection et au comptage de tremblements de terre, mais à la mesure de certaines caractéristiques d'évènements causés par les tremblements de terre, comme la vitesse des écoulements.

4.1.2 Couche physique pour le projet Clervolc

Pour le projet Clervolc, nous cherchons une couche physique adaptée aux nombreuses contraintes de l'environnement de déploiement, sans toutefois chercher à contribuer dans ce domaine directement.

4.1.2.1 Contraintes de l'environnement de déploiement

L'objectif du réseau est de surveiller des évènements causés par des tremblements de terre, comme des éboulements ou des écoulements chauds. Il est donc probable que les nœuds capteurs soient déplacés ou recouverts. Cela impose aux antennes utilisées d'être omnidirectionnelles.

Les nœuds capteurs seront déployés sur un volcan pendant une période de plusieurs mois, sans pouvoir bénéficier de maintenance. Afin d'éviter les effets de la malveillance humaine, les nœuds seront probablement dissimulés (ou camouflés dans de faux cailloux par exemple). Ils seront équipés d'antennes courtes, probablement au niveau du sol. Ces contraintes font que les nœuds capteurs ne seront pas en ligne de vue les uns des autres. Les signaux auront à traverser la végétation (la végétation au raz du sol, les feuilles tombées sur les antennes, le feuillage des arbres, etc.), la neige, les gouttes de pluie ou même la terre (si les nœuds sont enterrés suite à un éboulement). La qualité du lien radio en sera beaucoup réduite.

Ces contraintes inhabituelles sur le déploiement nous imposent à choisir une couche physique adaptée.

4.1.2.2 Choix de la couche physique

Afin de traverser la végétation (constituée d'une importante quantité d'eau) et la terre, l'utilisation de fréquences basses est conseillée. Les fréquences basses permettent aussi des transmissions à faible puissance. En revanche, elles nécessitent des

antennes de grande taille pour avoir un gain positif, et ne permettent que de faibles débits.

Tests de la bande de fréquences 2,4 GHz.

Dans [SPM06], les auteurs effectuent des mesures sur la bande de fréquences 2,4 GHz (avec des transmissions en DSSS et une modulation O-QPSK) lorsque l'un des nœuds capteurs est enterré. Les auteurs montrent que lorsque le nœud capteur n'est enterré qu'à 6 cm de profondeur, il n'est pas possible d'obtenir des communications fiables quand le nœud en surface est situé à plus de 7 m du nœud enterré.

Lors du stage de master recherche de Rezzag ABDERRAHMEN [Rez12], nous avons étudié les performances de la bande de fréquences 2,4 GHz (comme spécifiée dans la version de 2011 de la norme IEEE 802.15.4), au travers de la végétation (sèche). Les nœuds sont placés à environ 50 cm de hauteur dans une haie relativement dense. Un nœud émetteur, hors de la haie, émet 300 paquets sur une période de 5 minutes, à une puissance de 0 dBm. 9 nœuds récepteurs sont placés dans la haie, en ligne, avec un intervalle de 50 cm. D'après les tests, la puissance reçue chute rapidement de -45 dBm (pour le nœud récepteur situé à 50 cm) à -70 dBm (pour le nœud récepteur situé à 1,5 m). Ensuite, la puissance moyenne des paquets reçus diminue faiblement pour atteindre -76 dBm (pour le nœud récepteur situé à 4,5 m). Les tests donnent des taux moyens de paquets reçus très élevés (environ 95%) sur les distances utilisées (jusqu'à 4,5 m). Les résultats montrent que la puissance reçue dépend principalement de la profondeur d'enfouissement dans la végétation des nœuds émetteurs et récepteurs, plutôt que de la distance entre les nœuds. Plus globalement, ces tests laissent supposer que les communications opérant sur la bande de fréquences 2,4 GHz sont très pénalisées par la présence de végétation dense.

Tests des bandes de fréquences 868 MHz et 433 MHz.

Les deux bandes de fréquences 868 MHz et 433 MHz, faisant partie des bandes sans licence, semblent plus adaptées que la bande de fréquences 2,4 GHz pour les communications en présence de végétation ou lorsque les nœuds sont enterrés. En effet, les fréquences basses ont de nombreux avantages sur les fréquences hautes : les signaux se propagent plus loin, traversent mieux l'eau, et bénéficient davantage des effets de réflexion et de réfraction (ce qui permet une diffraction autour des obstacles).

La bande de fréquences 868 MHz est utilisée par la norme IEEE 802.15.4. Cette norme décrit trois couches physiques possibles : des transmissions en DSSS avec une modulation BPSK (opérant à 20 kbps), des transmissions en DSSS avec une modulation O-QPSK (présente de manière optionnelle dans la version de 2011 de la norme opérant à 100 kbps), et des transmissions en PSSS avec une modulation ASK (présente de manière optionnelle dans la version de 2011 de la norme, et opérant à 250 kbps).

La bande de fréquences autour de 433 MHz est envisagée par plusieurs chercheurs pour les communications souterraines. Dans [LVA07], les auteurs proposent un mo-

dèle de canal pour les communications souterraines et montrent que les fréquences proches de 400 MHz peuvent être utilisées pour les communications⁵. Dans [SV10], les auteurs caractérisent les communications entre nœuds enterrés et en surface, sur la bande de fréquences 433 MHz. Ils insistent sur le fait que l'antenne est un aspect principal du problème des communications souterraines. Quand un nœud émetteur est enfoui à 15 cm, un nœud récepteur en surface peut recevoir les trames jusqu'à 30 m, avec une antenne adaptée et quand le sol est sec.

4.1.3 Sous-couche MAC pour le projet Clervolc

Dans cette partie, nous cherchons à proposer un protocole MAC qui supporte la panne de nœuds, qui nécessite peu de configuration, et qui passe à l'échelle⁶. Ces contraintes nous conduisent à envisager une approche ne nécessitant pas de synchronisation entre les nœuds, mais garantissant des rendez-vous fréquents.

4.1.3.1 Étude des rendez-vous

Nous nous intéressons tout d'abord à un mécanisme de rendez-vous entre les nœuds [AGM13b*]. Ce mécanisme que nous proposons peut être vu comme une version non synchronisée de la sous-couche MAC en mode avec suivi de balises de IEEE 802.15.4.

La sous-couche MAC en mode avec suivi de balises de IEEE 802.15.4 synchronise les nœuds au moyen de balises, et utilise cette synchronisation pour mettre en place un rendez-vous commun à tous les nœuds, comme indiqué sur la figure 4.1. Cette approche, que nous nommons A_{sp} (pour approche synchronisée périodique), a l'avantage de garantir des rendez-vous longs entre tous les nœuds, c'est-à-dire de longs intervalles de temps pendant lesquels tous les nœuds sont actifs. Toutefois, elle présente de nombreux inconvénients : la contention pour le médium est grande car tous les nœuds sont actifs simultanément, et la synchronisation des nœuds est coûteuse (notamment lorsque le nombre de nœuds devient grand et que la topologie est multi-sauts).

L'approche A_{ap} (pour approche asynchronisée périodique) s'appuie sur l'approche A_{sp} , mais ne nécessite pas de synchronisation des nœuds : chaque nœud choisit l'instant de démarrage de son activité de manière indépendante. Cette approche est présentée sur la figure 4.2. Cette approche a l'avantage de ne pas nécessiter de synchronisation et de réduire la contention pour le médium (puisque peu de nœuds sont actifs simultanément), mais a plusieurs inconvénients concernant la connectivité : il peut exister des couples de nœuds qui ne peuvent pas communiquer directement (comme les couples (n_1, n_3) et (n_2, n_3) sur l'exemple de la figure), et les périodes d'activités communes entre couples peuvent être courtes.

L'approche A_{aa} (pour approche asynchronisée aperiodique) s'appuie sur l'approche A_{ap} , mais change l'instant de démarrage de l'activité de chaque nœud à

5. Les auteurs précisent notamment qu'ils vont utiliser la bande de fréquences 433 MHz pour effectuer leurs expérimentations.

6. Le passage à l'échelle vient du fait que le mécanisme proposé fonctionne indépendamment du nombre de nœuds dans le voisinage. Toutefois, les performances de la méthode d'accès dépendent évidemment du nombre de nœuds en compétition pour le médium.

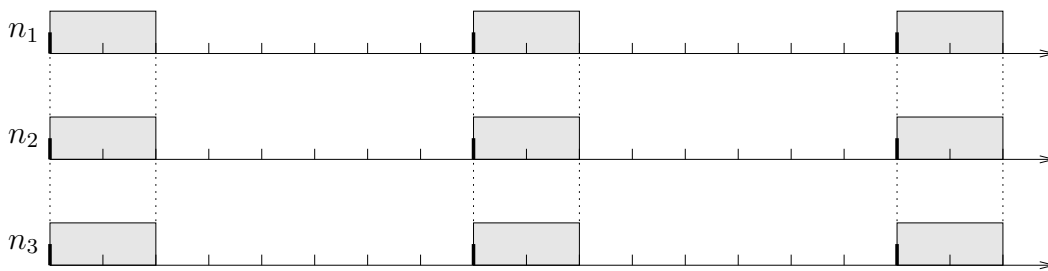


Figure 4.1 – Exemple d’activité de trois nœuds (au cours du temps) quand les cycles d’activité sont de 25% et que les activités sont synchronisées et périodiques (approche A_{sp}).

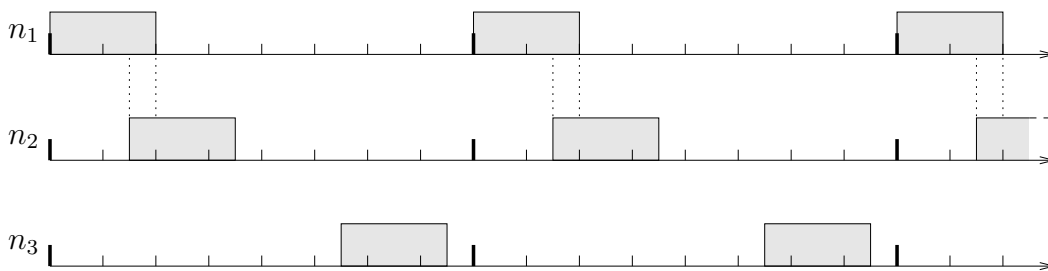


Figure 4.2 – Exemple d’activité de trois nœuds (au cours du temps) quand les cycles d’activité sont de 25% et que les activités ne sont pas synchronisées, mais sont périodiques (approche A_{ap}).

chaque cycle de manière aléatoire (avec un tirage uniforme), comme indiqué sur la figure 4.3. Cette approche a de nombreux avantages : elle ne nécessite pas de synchronisation, tous les couples de nœuds peuvent communiquer directement (quitte à attendre plusieurs cycles), les périodes d’activités communes peuvent être longues, et le nombre de nœuds actifs simultanément est généralement faible. L’inconvénient majeur de cette approche est qu’un rendez-vous entre deux nœuds peut parfois prendre un temps important.

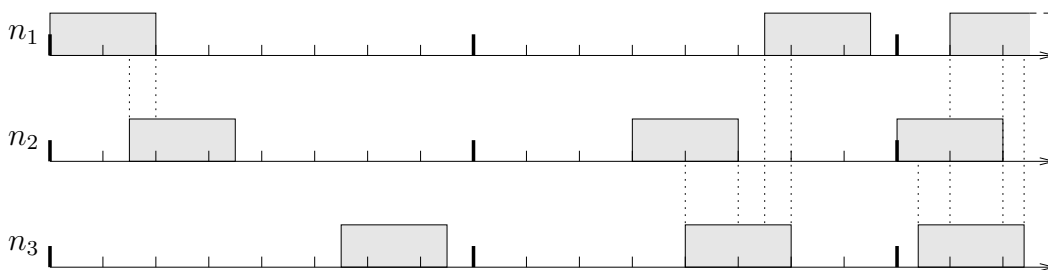


Figure 4.3 – Exemple d’activité de trois nœuds (au cours du temps) quand les cycles d’activité sont de 25% et que les activités ne sont ni synchronisées ni périodiques (approche A_{aa}).

Pour les approches A_{ap} et A_{aa} , les figures 4.2 et 4.3 (notamment) montrent des intervalles de temps synchronisés. Cette synchronisation sert à simplifier nos propos, et n’est pas une hypothèse forte du modèle. Si les nœuds sont vraiment désynchronisés, c’est-à-dire si la base de temps de chacun des trois nœuds (qui est modélisée

par les graduations sur les axes des figures de cette partie) est légèrement décalée d'un nœud à l'autre, seuls deux intervalles de temps seront inutilisables pour le rendez-vous : le premier intervalle de temps et le dernier intervalle de temps (quelque soit le nombre de nœuds concernés). Comme ces intervalles sont censés être petits (de l'ordre de $320 \mu s$), nous conservons dans la suite cette représentation synchronisée même si les approches ne le sont pas, et nous considérons que l'impact de la désynchronisation est négligeable (par rapport aux valeurs de t , de c et de $p.c$).

Dans la suite, nous comparons les performances de ces trois approches d'un point de vue analytique, selon deux métriques : la probabilité de rendez-vous productif entre deux nœuds et le délai moyen avant un tel rendez-vous. Il faut noter que le taux d'activité étant le même pour les trois approches, l'énergie moyenne dépensée est similaire.

Probabilité de rendez-vous productif.

Pour qu'un rendez-vous entre deux nœuds puisse être productif (et donc, correspondre à des échanges de trames), il faut que la durée du rendez-vous ne soit pas trop courte. Nous faisons l'hypothèse qu'il existe un seuil t à partir duquel un rendez-vous peut être considéré comme étant productif.

Nous allons chercher à évaluer analytiquement la probabilité $P_t^r(A_\alpha)$ de rendez-vous de deux nœuds pendant une durée ininterrompue supérieure ou égale à t , les nœuds fonctionnant selon l'approche A_α . Dans la suite, nous supposons que les instants de début d'activité des nœuds sont indépendants et identiquement distribués pour A_{ap} et A_{aa} . Notons n_1 et n_2 les deux nœuds considérés, c la durée du cycle et $p \in]0; 1]$ le taux d'activité des nœuds. Nous supposons enfin que $t \in]0; p.c]$.

Pour l'approche A_{sp} (qui nécessite une synchronisation de tous les nœuds), la probabilité de rendez-vous productif est simple à calculer. Pour $t \in]0; p.c]$, nous avons $P_t^r(A_{sp}) = 1$ puisque les nœuds se rencontrent systématiquement à chaque cycle pendant une durée $p.c$.

Pour l'approche A_{ap} , la probabilité de rendez-vous productif dépend des instants de début des nœuds n_1 et n_2 . Sans perte de généralité, on peut considérer que n_1 débute son activité en début de cycle. Trois cas peuvent se produire.

- n_2 démarre en même temps que n_1 , ou peu de temps après n_1 (au plus $p.c - t$ unités de temps), comme représenté sur la figure 4.4. Cette situation se produit avec une probabilité $(p.c - t)/c$ et garantit des rendez-vous pendant au moins t unités de temps. Elle conduit à $P_t^r(A_{ap}) = 1$.
- n_2 démarre peu de temps avant n_1 (au plus $p.c - t$ unités de temps), comme représenté sur la figure 4.5. Cette situation se produit avec une probabilité $(p.c - t - 1)/c$ et garantit des rendez-vous pendant au moins t unités de temps. Elle conduit à $P_t^r(A_{ap}) = 1$.
- n_2 démarre plus de $p.c - t$ unités de temps après n_1 , et plus de $p.c - t$ unités de temps avant la prochaine activité de n_1 . Cette situation se produit avec une probabilité $(c - 2.p.c + 2.t + 1)/c$ (obtenu par complémentarité) et empêche les rendez-vous pendant au moins t unités de temps. Elle conduit à $P_t^r(A_{ap}) = 0$.

Par sommation, on obtient : $P_t^r(A_{ap}) = (2p.c - 2t - 1)/c$.

Pour l'approche A_{aa} , la probabilité de rendez-vous productif par cycle de durée

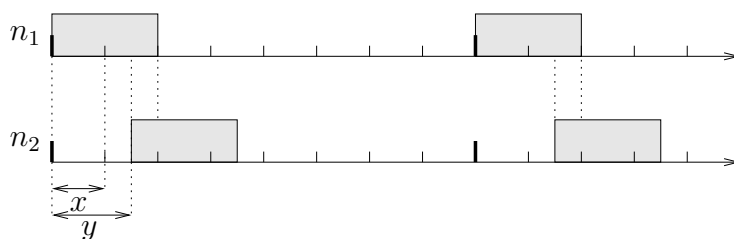


Figure 4.4 – Cas 1 : le nœud n_2 démarre peu de temps après le nœud n_1 . x représente un instant quelconque dans l'activité de n_1 , et y représente le retard de n_2 par rapport à n_1 .

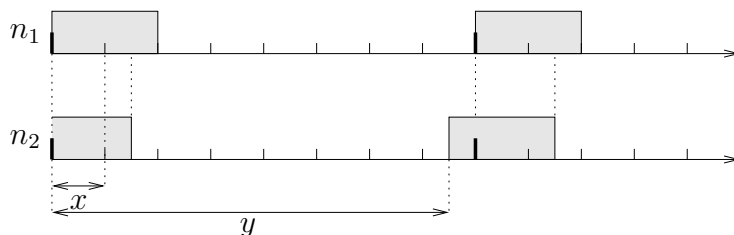


Figure 4.5 – Cas 2 : le nœud n_2 démarre peu de temps avant le nœud n_1 . x représente un instant quelconque dans l'activité de n_1 , et y représente l'avance de n_2 par rapport à n_1 .

c est égale à la probabilité de rendez-vous productif de A_{ap} , c'est-à-dire $P_t^{rc}(A_{aa}) = (2p.c - 2t - 1)/c$. La probabilité de rendez-vous productif à terme, $P_t^r(A_{aa})$, peut s'écrire de la manière suivante : $\sum_{i=1}^{+\infty} P_t^{rc}(A_{aa})(1 - P_t^{rc}(A_{aa}))^{i-1}$, i étant le numéro du cycle dans lequel le rendez-vous productif a lieu. Pour $t < p.c$, $P_t^{rc}(A_{aa}) \neq 0$, d'où $P_t^r(A_{aa}) = 1$. Pour $t = p.c$, $P_t^r(A_{aa}) = 1$ si l'on considère que les nœuds sont synchronisés sur les intervalles de temps, et $P_t^r(A_{aa}) = 0$ sinon.

Cette modélisation ne prend pas en compte la contention pour le canal (même si une partie de cette contention peut être intégrée dans le choix de la valeur de t). Toutefois, l'approche aléatoire du mécanisme permet à peu de nœuds en moyenne d'être actifs simultanément, ce qui est supposé réduire cette contention. L'étude par simulation du comportement de ce mécanisme est une perspective à court terme.

Délai moyen avant un rendez-vous productif.

À présent, nous allons chercher à évaluer analytiquement le délai moyen $d_t(A_\alpha)$ pour qu'un nœud n_1 , initialement actif, puisse avoir un rendez-vous productif avec un nœud n_2 donné, pendant une durée ininterrompue supérieure ou égale à t , ces deux nœuds fonctionnant selon l'approche A_α . Il faut noter que ce rendez-vous productif entre n_1 et n_2 n'a pas forcément lieu pendant la première activité de n_1 .

Pour l'approche A_{sp} , le délai moyen $d_t(A_{sp})$ peut s'exprimer de la manière suivante : $(\sum_{x=0}^{p.c-t-1} 0 + \sum_{x=p.c-t}^{p.c-1} (c-x)) / p.c$, $x \in [0; p.c[$ décrivant tous les instants possibles dans l'activité du nœud n_1 . En effet, pour tout instant $x \in [0; p.c - t[$, les nœuds n_1 et n_2 sont actifs et partagent encore au moins t unités de temps d'activité dans le cycle en cours : le délai avant le rendez-vous productif est donc

nul. En revanche, quand il reste moins de t unités de temps d'activité pour n_1 (ou pour n_2), il faut attendre la prochaine période d'activité, qui aura lieu après un délai de $c - x$ unités de temps. Après simplification, nous obtenons $d_t(A_{sp}) = (2.c - 2.p.c + t + 1).t / (2.p.c)$.

Pour l'approche A_{ap} , le délai moyen $d_t(A_{ap})$ est infini quand les nœuds n_1 et n_2 ne se rencontrent pas assez longtemps, ce qui se produit avec une probabilité de $1 - P_t^r(A_{ap})$. Quand les nœuds se rencontrent pendant au moins t unités de temps, le délai moyen est fonction du délai obtenu dans le cas où n_2 démarre peu de temps après n_1 (cas 1 de la figure 4.4) et du délai obtenu dans le cas où n_2 démarre peu de temps avant n_1 (cas 2 de la figure 4.5).

- Le délai moyen pour le cas 1 s'exprime comme la moyenne du délai pour tous les instants $x \in [0; p.c[$ de l'activité de n_1 , en prenant en compte tous les retards possibles (représentés par y) de l'activité de n_2 par rapport à celle de n_1 . Ce délai moyen d_1 s'exprime comme :

$$\sum_{y=0}^{p.c-t-1} \left(\sum_{x=0}^{y-1} (y-x) + \sum_{x=y}^{p.c-t-1} 0 + \sum_{x=p.c-t}^{p.c-1} (c+y-x) \right) / (p.c.(p.c-t)).$$

La somme pour x allant de 0 à $y-1$ prend en compte le délai avant le début de l'activité de n_2 . La somme pour x allant de y à $p.c-t-1$ correspond à une période où n_1 et n_2 sont actifs simultanément (et pendant au moins t unités de temps). La somme pour x allant de $p.c-t$ à $p.c-1$ prend en compte le délai avant la prochaine activité commune de n_1 et n_2 . On obtient $d_1 = (6c.t + p^2.c^2 - 5p.c.t + t^2 - 1) / (6p.c)$.

- Le délai moyen pour le cas 2 s'exprime comme la moyenne du délai pour tous les instants $x \in [0; p.c[$ de l'activité de n_1 , en prenant en compte toutes les avances possibles (représentées par y) de l'activité de n_2 par rapport à celle de n_1 . Ce délai moyen d_2 s'exprime comme :

$$\sum_{y=c-p.c+t+1}^{c-1} \left(\sum_{x=0}^{y+p.c-t-c} 0 + \sum_{x=y+p.c-t-c+1}^{p.c-1} (c-x) \right) / (p.c.(p.c-t+1)).$$

La somme pour x allant de 0 à $y+p.c-t-c$ correspond à une période où n_1 et n_2 sont actifs simultanément (pendant au moins t unités de temps). La somme pour x allant de $y+p.c-t-c+1$ à $p.c-1$ prend en compte le délai avant la prochaine activité commune de n_1 et n_2 . On obtient $d_2 = -(p.c-t-1)(-3c(p.c+t-2) + 2p^2.c^2 + 2p.c(t-2) - t(t-1)) / (6p.c(p.c-t+1))$.

Le délai moyen $d_t(A_{ap})$ est donc égal à $\alpha_1 d_1 + \alpha_2 d_2$, avec $\alpha_1 = (p.c-t) / (2p.c-2t-1)$, $\alpha_2 = (p.c-t-1) / (2p.c-2t-1)$, quand les nœuds n_1 et n_2 ont un rendez-vous suffisamment long (ce qui arrive avec une probabilité de $P_t^r(A_{ap})$).

Pour l'approche A_{aa} , le rendez-vous productif n'est pas possible si $t = p.c$. Pour $t < p.c$, le délai moyen $d_t(A_{aa})$ doit être étudié dans deux cas : le rendez-vous productif a lieu dans le premier cycle, ou le rendez-vous a lieu dans un cycle ultérieur.

- Le rendez-vous a lieu pendant le premier cycle avec une probabilité $P_t^r(A_{ap})$. Si l'activité de n_1 démarre avant celle de n_2 (ce qui se produit avec une probabilité de $(p.c-t) / (2p.c-2t-1)$), le délai moyen est :

$$\sum_{y=0}^{p.c-t} \left(\sum_{x=0}^{y-1} (y-x) + \sum_{x=y}^{p.c-t-1} 0 \right) / ((p.c-t)(p.c-t)),$$

en tirant parti du fait que le rendez-vous a lieu durant le cycle. Si l'activité de n_2 démarre avant celle de n_1 (ce qui se produit avec une probabilité de $(p.c-t-1) / (2p.c-2t-1)$), le délai moyen est :

$$\sum_{y=c-p.c+t+1}^{c-1} 0 / ((p.c-t)(p.c-t+1)),$$

à nouveau en tirant parti du fait que le rendez-vous a lieu durant le cycle.

Globalement, on obtient $d_1 = (p.c - t + 1)(p.c - t + 2)/(6(p.c - t - 1))$.

- Le rendez-vous a lieu pendant un cycle ultérieur avec une probabilité $1 - P_t^r(A_{ap})$. Dans ce cas, si x parcourt l'ensemble des instants dans l'activité de n_1 du premier cycle, on a :

$$d_2(x) = (c - x) + \sum_{i=0}^{+\infty} (p(1-p)^i c.i) + (d_1 + (\sum_{z=0}^{c-p.c} z)/(c - pc)),$$

où $p = P_t^r(A_{aa})$ est la probabilité que les nœuds se rencontrent en un cycle donné (pendant au moins t unités de temps). La première partie du délai correspond au temps entre l'instant x et le début du deuxième cycle. La deuxième partie du délai correspond à la durée moyenne de l'attente entre le début du deuxième cycle et le début du cycle où n_1 et n_2 se rencontrent. La troisième partie du délai correspond à la durée entre le début du cycle où n_1 et n_2 se rencontrent et le début de l'activité de n_1 , puis entre le début de l'activité de n_1 et le début de la rencontre de n_1 et de n_2 . On obtient $d_2(x) = (c - x) + c(1-p)/p + d_1 + c(1-p)(c(1-p) + 1)/(2c(1-p))$. Finalement, on a $d_2 = \sum_{x=0}^{p.c-1} d_2(x)/(p.c) = (1 - c(p-2))/2 + c(1-p)/p + d_1 + c(1-p)(c(1-p) + 1)/(2c(1-p)) = (1-p)c/p + ((1-p)c + 1)/2 + (1 - (p-2)c)/2 + d_1$.

Le délai $d_t(A_{aa})$ s'exprime donc de la manière suivante : $d_t(A_{aa}) = P_t^r(A_{aa}) \cdot d_1 + (1 - P_t^r(A_{aa})) \cdot d_2$.

Le calcul du délai $d_t(A_{aa})$ nous permet donc de quantifier (en partie) la réactivité d'un protocole qui serait basé sur des rendez-vous aveugles (c'est-à-dire sans connaissance *a priori*), en fonction des paramètres comme la durée du cycle et le taux d'activité des nœuds.

4.1.3.2 Protocole MAC qui passe à l'échelle

L'approche A_{aa} permet à des nœuds de communiquer sans besoin de synchronisation, et réalise ainsi des rendez-vous aveugles, ce qui est assez rare dans les protocoles MAC de la littérature [MMKR13]. Le délai avant les rendez-vous productifs est généralement raisonnable : pour un taux d'activité de 25%, les nœuds attendent en moyenne moins d'un cycle ; pour un taux d'activité de 15% ou plus, les nœuds attendent en moyenne moins de deux cycles. Dans cette partie, nous proposons des pistes pour construire un protocole MAC basé sur cette approche A_{aa} .

Le protocole MAC que nous envisageons tourne autour des concepts suivants. Tous les nœuds du réseau partagent une durée de cycle commune *a priori*. Le comportement de chaque nœud suit un automate où chaque état correspond aux besoins de surveillance (qui dépend de l'importance des événements détectés). Par exemple, un nœud peut être dans un état d'attente (si aucun événement significatif n'est détecté) ou dans un état d'alertes (si un événement détecté déclenche une surveillance intensive). L'état indique au nœud son taux d'activité, et a une influence sur les paramètres réseaux. L'automate indique les changements d'état des nœuds en fonction des événements détectés, des messages reçus ou du temps écoulé.

L'accès au médium se fait de la manière suivante. Pour respecter l'approche A_{aa} , chaque nœud démarre son activité de manière indépendante, et à chaque cycle, décide aléatoirement de sa période d'activité au sein du cycle. Pendant l'activité d'un nœud, l'accès au médium se fait selon le mécanisme CSMA/CA non slotté. En début d'activité, chaque nœud envoie une balise en diffusion (en CSMA/CA non slotté)

pour prévenir ses voisins actifs. Cette balise est prioritaire par rapport aux envois de données du nœud, et ne peut pas être détruite en cas d'échec d'accès au médium (sauf en fin de période d'activité du nœud). Cette balise contient l'identification de l'état du nœud et un numéro de séquence.

Le changement d'état est réalisé de la manière suivante. Pour un nœud n , notons l_n son état, p_n son taux d'activité et s_n le numéro de séquence actuel. Lorsqu'un évènement nécessitant un changement d'état est détecté par un nœud, l'état du nœud et son taux d'activité changent (pour s'adapter à la nouvelle situation) et le numéro de séquence que le nœud utilise est incrémenté. Lorsqu'un nœud n reçoit une balise d'un voisin v , n adapte son état en fonction de celui de v . Pour cela, n réalise les opérations suivantes. Si $s_v > s_n$, le paquet venant de v contient des informations plus à jour que celles de n , et n met donc à jour son état : s_n prend la valeur de s_v , l_n prend la valeur de l_v , et p_n prend la valeur de p_v . Si $s_v = s_n$, les deux cas suivants peuvent se produire : soit le paquet venant de v contient les mêmes informations que celles connues de n , soit le changement d'état qui a causé l'augmentation de s_v est indépendant de celui qui a causé l'augmentation de s_n . Pour résoudre l'incohérence survenant dans ce deuxième cas, nous choisissons de fixer l'état de n à p_n ou p_v , en fonction de l'état qui correspond au plus grand taux d'activité. l_n prend la valeur de l'état l_n ou l_v correspondant au plus grand taux d'activité.

Le choix du temps de cycle, noté c , est un paramètre important du protocole. Considérons un temps de cycle de base c et un taux d'activité p donné. Notons $c_i = c/2^i$, avec $i \geq 0$. Pour chaque i , le même taux d'activité p peut être obtenu en ayant une période d'activité d'une durée de $p.c_i$ toutes les c_i unités de temps. L'augmentation de i conduit à une réduction du délai avant le prochain rendez-vous (ce qui est un avantage pour le mécanisme de rendez-vous), mais aussi à une réduction de la durée de rendez-vous (ce qui est un inconvénient pour le protocole car cela réduit la quantité de trafic échangeable pendant un rendez-vous) [AGM13a*]. Ainsi, i ne peut pas prendre des valeurs arbitrairement grandes, car la durée de rendez-vous doit être suffisamment grande pour que deux nœuds qui se rencontrent puissent échanger des informations⁷. Nous suggérons d'utiliser la plus grande valeur de i qui garantit une durée moyenne de rendez-vous dépassant un seuil t permettant un échange d'informations. Une valeur par défaut de t peut être fixée à 15,36 ms (ce qui correspond à la plus petite taille de supertrame dans IEEE 802.15.4 en mode avec suivi de balises).

Nous faisons l'hypothèse que le protocole MAC du projet Clervolc n'utilise (dans un premier temps) que deux états : un état d'attente (utilisé quand aucun évènement significatif n'est détecté, et correspondant à un taux d'activité de 10% ou moins) et un état d'alertes (utilisé quand un évènement significatif est détecté, et correspondant à un taux d'activité de 50%). La durée du cycle c est fixée à une seconde. La valeur de i garantissant une durée moyenne de rendez-vous légèrement supérieure à 15,36 ms n'est pas encore déterminée pour notre protocole, car elle dépend à la fois du taux d'activité choisi et de la surcharge protocolaire. Ces paramètres sont

7. Cette durée doit être d'autant plus grande que du temps est perdu en début d'activité commune (à cause de l'envoi des balises en CSMA/CA non slotté, ce qui prend un temps aléatoire, même en l'absence de collisions) et en fin d'activité commune (quand il ne reste plus assez de temps pour envoyer une trame complète en CSMA/CA non slotté).

actuellement en cours d'étude dans le cadre de la thèse d'Affoua Thérèse ABY, dans la perspective d'être obtenus par simulation (et éventuellement par prototypage, si le protocole est retenu).

4.1.4 Couche réseau pour le projet Clervolc

La couche réseau envisagée pour le projet Clervolc est composée de deux protocoles de routage : un protocole d'attente et un protocole d'alertes. Ces deux protocoles se basent sur des rendez-vous aveugles, avec un paramétrage différent.

4.1.4.1 Le protocole d'attente

Le protocole d'attente concerne la surveillance périodique non critique. Ce protocole est utilisé en combinaison avec l'état d'attente (très économe en énergie) du protocole MAC du projet Clervolc, pour lequel les nœuds sont rarement éveillés. Les nœuds étant désynchronisés et souvent en sommeil (pour économiser l'énergie), ils ont peu d'occasions de communiquer, ce qui fait que les liens de la topologie sont peu souvent disponibles. La topologie résultante est donc très dynamique.

Les approches de routage par gradient sont adaptées aux réseaux dynamiques, étant donné qu'elles génèrent une faible surcharge. Les approches par gradient fonctionnent de la manière suivante. Pour chaque destination potentielle d , chaque nœud n stocke le coût $c(n, d)$ pour atteindre la destination. Ce coût est annoncé périodiquement (lorsque c'est possible) aux voisins immédiats du nœud. Lorsqu'un nœud n reçoit l'annonce du coût de son voisin v , n réalise l'opération suivante : si $c(v, d) + \alpha < c(n, d)$, alors n met à jour son coût en $c(v, d) + \alpha$, α représentant le coût (généralement égal à 1) de la transmission de n à v . Lorsque n a un paquet à acheminer à d , n le transmet à n'importe lequel de ses voisins v tel que $c(v, d) < c(n, d)$.

Nous envisageons d'utiliser pour le protocole d'attente une telle approche par gradient. Comme les nœuds sont rarement éveillés, il nous semble important que le coût intègre des aspects temporels : la fonction de coût $c(n, d)$ pourrait combiner la distance $dist(n, d)$ à la destination d et le délai $delay(n, d)$ depuis la dernière rencontre d'un voisin permettant d'acheminer la trame. Le coût $c(n, d)$ serait par exemple égal à $dist(n, d)$ si $delay(n, d) \leq delay_{max}$, et à $dist(n, d) + \beta \cdot (delay(n, d) - delay_{max})$ sinon, $delay_{max}$ étant le délai maximum toléré pour le rafraîchissement des entrées de voisinage, et β étant une constante permettant de pondérer le délai au delà de $delay_{max}$. L'utilisation de la borne $delay_{max}$ est nécessaire pour prendre en compte les changements (supposés rares) de topologie. Si $delay(n, d) > delay_{max}$, on aurait aussi pu choisir de fixer $c(n, d)$ à une valeur arbitrairement grande.

4.1.4.2 Le protocole d'alertes

Le protocole d'alertes concerne la surveillance lorsqu'un évènement significatif est détecté. Il est utilisé conjointement à un protocole MAC très réactif, pour lequel les nœuds sont souvent éveillés : il s'agit du protocole MAC du projet Clervolc ayant un taux d'activité élevé.

Le protocole d'alertes que nous envisageons est le protocole AODV⁸. L'avantage d'AODV dans notre contexte est qu'il passe à l'échelle étant donné qu'il ne nécessite pas de connaissance complète de la topologie, et qu'il ne construit une route que pour la destination des alarmes. Nous supposons que l'inondation du réseau pour établir une route avant d'envoyer les alarmes n'est pas un inconvénient majeur, puisqu'en cas d'alertes, les nœuds sont supposés être actifs la plupart du temps⁹. De plus, AODV devrait être en mesure de gérer la dynamique de la topologie pendant la période de transition entre l'état d'attente et l'état d'alertes (cette dynamique n'étant pas dû à la mobilité des nœuds, mais à l'intermittence des liens entre nœuds), puisqu'il intègre des mécanismes locaux de réparation des liens.

4.1.4.3 Étude de la combinaison des protocoles

Dans cette partie, nous allons montrer que nos protocoles d'attente et d'alertes ne sont pas compatibles, et qu'ils peuvent donc conduire à des boucles pendant la période de transition de l'un à l'autre.

La figure 4.6 présente un exemple de réseau de six nœuds, où a est la destination. À cause de rares rendez-vous productifs entre b et d , le nœud d peut considérer que b n'est plus joignable, et sélectionner f comme prochain saut pour aller vers a . Cette décision se produit quand la durée depuis la dernière rencontre entre b et d dépasse $delay_{max}$, et que le surcoût lié à ce dépassement de délai est supérieur au surcoût en terme de distance de f à a . Lors de la transition d'un protocole de routage à l'autre, une boucle peut apparaître si d envoie les paquets à f (selon le protocole d'attente \mathcal{R}_1), et si f envoie les paquets à d (selon le protocole d'alertes \mathcal{R}_2). En d'autres termes, même si les deux protocoles utilisent une notion de coût basée sur la distance (ce qui pourrait laisser supposer que les protocoles sont compatibles), la dynamique de la topologie pour le protocole d'attente ne permet pas d'avoir la propriété de compatibilité recherchée.

4.1.5 Bilan concernant le projet Clervolc

Le projet Clervolc est l'un des projets de l'équipe qui concerne la thématique environnementale. Il s'agit d'ailleurs du projet le plus important de cette thématique, à la fois par rapport à son envergure et aux points abordés. Il s'agit d'un projet récent, pour lequel les résultats sont encore prospectifs. Lors du stage de master recherche de Rezzag ABDERRAHMEN, nous avons effectué des mesures préliminaires pour étudier le comportement de la bande de fréquences 2,4 GHz au travers de végétation dense. Actuellement, pendant la thèse d'Affoua Thérèse ABY, nous étudions des protocoles pouvant être utilisés dans le contexte du projet Clervolc.

8. Nous envisageons aussi d'utiliser pour protocole d'alertes le protocole Pirat [ERGBM09*], qui est basé sur AODV et qui a été conçu spécifiquement pour gérer les alarmes. Toutefois, Pirat a été développé sur la base d'une couche MAC active en continu, ce qui n'est pas forcément le cas dans le projet Clervolc.

9. Des résultats préliminaires que nous avons obtenus dans [ERGBM09*] montrent toutefois que pour un réseau d'une centaine de nœuds déployés en grille, l'établissement de routes par AODV peut prendre environ 9 secondes. Il sera probablement nécessaire d'utiliser un paramétrage spécifique des paramètres de AODV pour que ce protocole puisse être utilisé dans notre contexte.

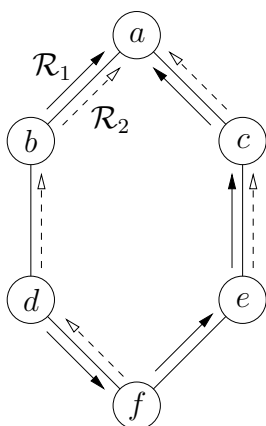


Figure 4.6 – Le protocole d’attente et le protocole d’alertes ne sont pas compatibles. En effet, à cause des rares rencontres, le nœud d peut déterminer que son prochain saut pour a selon le protocole d’attente (indiqué par \mathcal{R}_1) est f , qu’il vient de rencontrer, plutôt que b , qu’il n’a pas rencontré récemment. Le protocole d’alertes (indiqué par \mathcal{R}_2), envoie les paquets de f à d , qui est sur le plus court chemin vers a , causant ainsi une boucle.

Dans les perspectives à court terme concernant le projet Clervolc, nous devons proposer des protocoles MAC et réseau adaptés aux événements à surveiller. Il s’agit donc pour nous d’affiner les protocoles que nous avons conçus, de les simuler, et de les tester en grandeur nature sur un site représentatif (qui pourrait tout d’abord être le Puy de Dôme, à Clermont-Ferrand, puis le Merapi, en Indonésie, dans les deux cas avec l’aide de nos collaborateurs du LMV). Pour faciliter les travaux autour du matériel (qui doit entre autres être tropicalisé pour résister aux intempéries et à l’environnement particulier d’un volcan actif) et pour les tâches de déploiement sur sites, j’encadre Loïc LE GUENNEC, un ingénieur nouvellement recruté dans le cadre du projet Clervolc pour travailler dans l’équipe et avec le LMV pendant deux ans.

4.2 Architectures multi-piles globales

Nous allons à présent introduire un nouveau type d’architectures multi-piles : les architectures globales.

Définition 13 (Architecture globale). Une architecture multi-piles est dite globale quand, pour chaque paquet, plusieurs piles protocolaires peuvent être utilisées pour l’acheminement de bout en bout, et quand la décision d’utiliser une pile ou une autre dépend d’un contexte non exclusivement local au nœud qui achemine. Ce contexte global peut inclure la détection d’un événement par un nœud du réseau (ce nœud étant généralement distant du nœud qui achemine le paquet), ou la transmission d’une commande par un opérateur (*via* un nœud passerelle distant), par exemple.

Les architectures multi-piles globales sont utilisées lorsque la décision d’utiliser une pile ou une autre n’est pas prise par chaque nœud indépendamment, mais est prise soit de manière collective, soit par un nœud distant. Par exemple, dans le

cadre du projet Clervolc, la prise de décision est prise par un nœud distant : soit le puits (qui s'appuie sur l'ensemble des mesures rapatriées par les nœuds, ou sur une décision humaine d'un opérateur), soit un nœud détectant un évènement volcanique à surveiller (il peut alors s'agir d'un nœud proche du sommet). Dans ces deux cas, la décision prise par le nœud a un impact sur l'ensemble du réseau, puisqu'elle induit un changement de protocoles (ou de paramétrage des protocoles) pour tous les nœuds.

Dans la suite, nous décrivons la notion d'état d'un nœud, puis d'état du réseau. Nous donnons ensuite des propriétés pour les protocoles qui supportent le changement d'états.

4.2.1 États d'un nœud

L'état d'un nœud définit la vision qu'il a de l'état du réseau et de l'environnement surveillé. L'état d'un nœud a une influence directe sur la pile protocolaire qu'il utilise (et éventuellement sur les applications qu'il exécute).

Le nœud change d'état en fonction d'évènements extérieurs qu'il détecte (ou qu'il cesse de détecter), ou en fonction de messages qu'il reçoit. Les états d'un nœud, et les transitions entre ces états, peuvent être représentées sous la forme d'un automate. Cet automate est défini au préalable par l'opérateur du réseau, et est implémenté dans les nœuds. On peut envisager plusieurs types d'automates (décrits du plus simple au plus compliqué).

- Un automate linéaire à deux états, comme représenté sur la figure 4.7. Un exemple est la surveillance d'un évènement unique. Le nœud démarre dans un état d'attente (l'état A), qui correspond à une très faible consommation d'énergie. Lorsque le nœud détecte un évènement significatif, il bascule dans un état d'alertes (l'état B), où la surveillance devient intensive. Le nœud reste indéfiniment dans cet état d'alertes.
- Un automate linéaire à $n \geq 3$ états, comme représenté sur la figure 4.8. Un exemple est la surveillance d'évènements qui se produisent de manière successive, ou la surveillance d'évènements de plus en plus importants.
- Un automate arborescent, comme représenté sur la figure 4.9. Un exemple est la surveillance de plusieurs types d'évènements exclusifs (correspondant aux états B et C).
- Un automate avec cycles, comme représenté sur la figure 4.10. Un exemple est la surveillance d'évènements multiples et discontinus, qui nécessite d'alterner entre un état d'attente et un état de travail. Un autre exemple est la surveillance d'évènements avec erreurs de détection, où un nœud peut détecter *a posteriori* que la décision de changement d'état qu'il avait faite était erronée, et ainsi revenir sur cette décision.

Dans une application typique, l'automate des états d'un nœud est une combinaison de ces types d'automates.

4.2.2 États du réseau

L'évolution des états des nœuds doit être coordonnée, afin que l'état global du réseau soit cohérent. Nous imposons donc qu'à chaque instant, tous les nœuds du



Figure 4.7 – Un automate linéaire à deux états : l'état A peut être un état d'attente, et l'état B un état de travail (permanent).

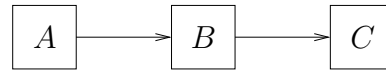


Figure 4.8 – Un automate linéaire à trois états : l'état A peut être un état d'attente, l'état B un état de travail modéré, et l'état C un état de travail intensif.

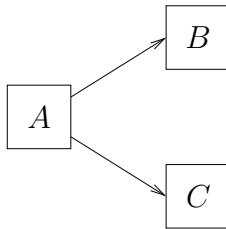


Figure 4.9 – Un automate arborescent : l'état A peut être un état d'attente, l'état B un état de surveillance d'un type d'évènement, et l'état C un état de surveillance d'un autre type d'évènement.

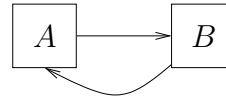


Figure 4.10 – Un automate avec cycles : l'état A peut être un état d'attente, et l'état B un état de travail (temporaire).

réseau soient dans deux états au plus¹⁰. Plus formellement, si l'on note E_t l'ensemble des états de chacun des nœuds à un instant t , nous imposons que pour tout t , $|E_t| \leq 2$. Cela signifie que lorsque $|E_t| = 2$, les nœuds du réseau évoluent vers un état commun (atteint pour un certain t' avec $t < t'$ et $|E_{t'}| = 1$). De même, lorsque $|E_t| = 2$, les nœuds du réseau ont évolué à partir d'un état commun (atteint pour un certain t'' avec $t'' < t$ et $|E_{t''}| = 1$). Dans la suite, l'état commun du passé est noté état 1, et l'état commun du futur est noté état 2.

L'état d'un nœud ne peut pas changer à plusieurs reprises dans un délai trop court : chaque nœud doit attendre que tous les autres nœuds du réseau aient atteint le même état avant de changer à nouveau d'état, afin que l'état du réseau puisse être cohérent¹¹. Nous ajoutons donc une contrainte aux transitions de l'automate caractérisant l'état d'un nœud : le temps minimum de passage d'un état à un autre est borné par un délai dépendant de la topologie notamment. Nous calculons ce temps minimum dans la suite, en 4.3.2.3.

Le changement d'état d'un nœud implique un changement de protocoles MAC et réseau. Cela signifie qu'à un instant donné, certains nœuds utilisent un protocole \mathcal{M}_1 pour accéder au médium alors que d'autres utilisent un protocole \mathcal{M}_2 . Pour que les nœuds utilisant \mathcal{M}_1 puissent communiquer avec les nœuds utilisant \mathcal{M}_2 , il faut que \mathcal{M}_1 et \mathcal{M}_2 puissent permettre aux nœuds d'interagir malgré les différences de méthodes d'accès.

10. Dans l'annexe A.1, nous considérons aussi des évolutions multi-états, où les nœuds du réseau peuvent être dans plus de deux niveaux. Nous supposons simplement que ces états peuvent être ordonnés (par ordre d'importance, par exemple).

11. Si deux évènements produisant des états différents se produisent dans un délai court, un état commun doit être choisi pour tous les nœuds. Nous supposons dans ce cas que ces états sont compatibles, c'est-à-dire qu'il existe dans l'automate des états un état qui suit ces deux états.

4.2.3 Protocoles qui supportent le changement d'état

Dans cette partie, nous étudions la manière dont les protocoles des couches MAC et réseau peuvent supporter le changement d'état. Pour cela, nous introduisons la notion de p -rejoignabilité, qui indique si des nœuds fonctionnant selon des protocoles MAC différents peuvent interagir ou non.

4.2.3.1 Étude de la p -rejoignabilité MAC

Pour qu'un nœud a , exécutant un protocole MAC \mathcal{M}_1 , puisse échanger des informations avec un nœud b , exécutant un protocole MAC \mathcal{M}_2 , il faut entre autres que a et b aient des périodes d'activité communes, et que les formats de trames de \mathcal{M}_1 et de \mathcal{M}_2 soient compatibles.

Définition 14 (p -rejoignabilité). Soit \mathcal{M}_1 et \mathcal{M}_2 deux protocoles MAC. Soit a un nœud quelconque exécutant \mathcal{M}_1 et b un nœud quelconque exécutant \mathcal{M}_2 , tels que a et b sont à portée. Les deux protocoles \mathcal{M}_1 et \mathcal{M}_2 sont p -rejoignables si la probabilité que a et b puissent communiquer est p .

Les protocoles MAC de IEEE 802.15.4, en mode avec ou sans suivi de balises, ont des formats de trames compatibles. Pour deux protocoles MAC \mathcal{M}_1 et \mathcal{M}_2 de cette famille, \mathcal{M}_1 et \mathcal{M}_2 sont p -rejoignables, avec une probabilité p qui dépend de nombreux paramètres : le taux d'activité des nœuds a et b (si l'un de ces nœuds est un FFD et que le protocole correspondant est IEEE 802.15.4 en mode sans suivi de balises, le taux d'activité de ce nœud est de 100%), l'instant d'envoi des balises et la probabilité que l'un des nœuds cause une collision avec la balise de l'autre¹².

Si \mathcal{M}_1 et \mathcal{M}_2 correspondent tous les deux au protocole MAC proposé pour le projet Clervolc [AGM13b*], avec les taux d'activité p_1 et p_2 respectivement, alors ces deux protocoles sont 1-rejoignables.

La définition de p -rejoignabilité considère un temps t minimum pendant lequel deux nœuds peuvent communiquer : en pratique, les nœuds ne peuvent pas utiliser l'ensemble de leur temps d'activité commun pour communiquer, en raison (notamment) de la contention pour le médium dans le voisinage de ces nœuds.

4.2.3.2 Protocoles MAC de transition

Lorsque l'application nécessite que le réseau passe d'un protocole MAC à un autre, le critère de p -rejoignabilité peut être utilisé pour connaître la probabilité que la transition se produise (et, par conséquent, le délai de transition).

Nous cherchons à présent à déterminer si, dans certains cas, passer d'un protocole \mathcal{M}_1 à un protocole \mathcal{M}_3 via un protocole \mathcal{M}_2 intermédiaire n'est pas plus efficace

12. Comme CSMA/CA non slotté, utilisé par IEEE 802.15.4 en mode sans suivi de balises, ne nécessite qu'un seul CCA avant une transmission, alors que CSMA/CA slotté, utilisé par IEEE 802.15.4 en mode avec suivi de balises, en nécessite deux, il est possible qu'un nœud exécutant CSMA/CA non slotté soit favorisé pour ses transmissions par rapport à un nœud exécutant CSMA/CA slotté. Toutefois, l'avantage de ce nœud n'apparaît que lorsque les frontières des intervalles de temps de CSMA/CA slotté et de CSMA/CA non slotté sont éloignées d'au plus 16 μs (les intervalles faisant 320 μs). En effet, la couche physique de IEEE 802.15.4 effectue 8 mesures de 16 μs chacune par demande de CCA.

que de passer directement de \mathcal{M}_1 à \mathcal{M}_3 . Si c'est le cas, le protocole intermédiaire \mathcal{M}_2 , sans intérêt applicatif particulier, peut être utilisé uniquement pour faciliter le passage de \mathcal{M}_1 à \mathcal{M}_3 . Dans ce cas, dans l'automate des nœuds, le protocole \mathcal{M}_2 correspond alors à un état virtuel.

Dans la suite, nous considérons trois protocoles MAC : \mathcal{M}_1 , \mathcal{M}_2 et \mathcal{M}_3 . Nous faisons l'hypothèse que les activités des protocoles sont indépendantes (comme c'est le cas dans le protocole MAC que nous envisageons pour le projet Clervolc). \mathcal{M}_i et \mathcal{M}_j sont $p_{i,j}$ -rejoignables, pour i et j dans $\{1, 2, 3\}$. Nous cherchons à exprimer $p_{1,3}$ en fonction de $p_{1,2}$ et de $p_{2,3}$. Rappelons que l'utilisation du protocole intermédiaire \mathcal{M}_2 est avantageuse quand la probabilité $p_{1,3}$ est faible mais que $p_{1,2}$ et $p_{2,3}$ sont élevés.

La figure 4.11 présente la p -rejoignabilité des protocoles \mathcal{M}_1 et \mathcal{M}_3 , en passant ou non par un protocole intermédiaire \mathcal{M}_2 , pour un jeu de valeurs données. Nous notons p_i le taux d'activité du protocole \mathcal{M}_i (pour $i \in \{1, 2, 3\}$). Sur l'exemple de la figure, nous avons $p_{1,2} = 10\%$ et $p_{2,3} = 40\%$. On peut déduire que le taux d'activité p_2 du protocole \mathcal{M}_2 est tel que $40 \leq p_2 \leq 100$.

- Quand $p_2 = 100\%$, le taux d'activité p_1 est obligatoirement égal à 10% (puisque $p_{1,2} = 10\%$). De même, le taux d'activité p_3 est obligatoirement égal à 40% (puisque $p_{2,3} = 40\%$). La probabilité $p_{1,3}$ est comprise entre 0% (cas défavorable où les 10% d'activité de \mathcal{M}_1 sont disjoints des 40% d'activité de \mathcal{M}_2) et 10% (cas favorable où les 10% d'activité de \mathcal{M}_1 sont communs avec une partie des 40% d'activité de \mathcal{M}_3).
- Quand $p_2 = 70\%$, le taux d'activité p_1 est compris entre 10% (cas où \mathcal{M}_1 n'est actif qu'en même temps que \mathcal{M}_2) et 40% (cas où \mathcal{M}_1 est actif pendant $p_{1,2} = 10\%$ du temps avec \mathcal{M}_2 , et pendant les $100 - p_2 = 30\%$ du temps où \mathcal{M}_2 est inactif). De même, le taux d'activité p_3 est compris entre 40% et 70% (cas où \mathcal{M}_3 est actif pendant $p_{2,3} = 40\%$ du temps avec \mathcal{M}_2 , et pendant les $100 - p_2 = 30\%$ du temps où \mathcal{M}_2 est inactif). La probabilité $p_{1,3}$ est donc comprise entre 0% et 40% (cas où \mathcal{M}_1 et \mathcal{M}_3 sont tous les deux actifs pendant 40% du temps, et que leurs activités sont communes).
- Quand $p_2 = 40\%$ (ce qui correspond au minimum d'activité de \mathcal{M}_2 étant donné que $p_{2,3} = 40\%$), le taux d'activité p_1 est compris entre 10% et 70% . De même, le taux d'activité p_3 est compris entre 40% et 100% . La probabilité $p_{1,3}$ est donc comprise entre 0% et 70% .

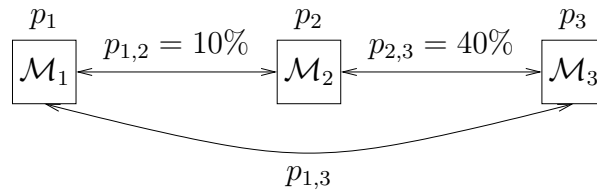


Figure 4.11 – Transitions d'un protocole MAC \mathcal{M}_1 à un protocole MAC \mathcal{M}_3 , en passant ou non par un protocole MAC intermédiaire \mathcal{M}_2 . Les valeurs de $p_{1,2}$ et $p_{2,3}$ sont données à titre d'exemple.

Les propriétés suivantes formalisent la démarche que nous venons d'utiliser. Elles permettent de borner p_2 , puis p_1 et p_3 , et finalement permettent d'en déduire un

intervalle possible pour $p_{1,3}$.

Propriété 11. Soit \mathcal{M}_1 , \mathcal{M}_2 et \mathcal{M}_3 trois protocoles MAC, tels que \mathcal{M}_i et \mathcal{M}_j sont $p_{i,j}$ -rejoignables (avec $i < j$). Alors, $\max\{p_{1,2}, p_{2,3}\} \leq p_2$.

Démonstration. Comme $p_{1,2}$ correspond à une activité commune de \mathcal{M}_1 et \mathcal{M}_2 , on a $p_{1,2} \leq p_2$. De même, $p_{2,3} \leq p_2$. On en déduit la propriété. \square

Propriété 12. Soit \mathcal{M}_1 , \mathcal{M}_2 et \mathcal{M}_3 trois protocoles MAC, tels que \mathcal{M}_i et \mathcal{M}_j sont $p_{i,j}$ -rejoignables (avec $i < j$). Alors, $p_1 \in [p_{1,2}; \min\{1, p_{1,2} + 1 - p_2\}]$ et $p_3 \in [p_{2,3}; \min\{1, p_{2,3} + 1 - p_2\}]$.

Démonstration. On a $p_1 \geq p_{1,2}$ car $p_{1,2}$ correspond à une activité commune de \mathcal{M}_1 et \mathcal{M}_2 . Dans le cas le plus favorable, \mathcal{M}_1 est actif en même temps que \mathcal{M}_2 avec un taux $p_{1,2}$, et \mathcal{M}_1 est actif pendant que \mathcal{M}_2 est inactif avec un taux d'au plus $1 - p_2$. On a bien $p_1 \leq p_{1,2} + 1 - p_2$. La preuve est similaire pour p_3 . \square

Propriété 13. Soit \mathcal{M}_1 , \mathcal{M}_2 et \mathcal{M}_3 trois protocoles MAC, tels que \mathcal{M}_i et \mathcal{M}_j sont $p_{i,j}$ -rejoignables (avec $i < j$). Alors, $\max\{0, p_{1,2} + p_{2,3} - 1\} \leq p_{1,3} \leq \min\{p_1, p_3\}$.

Démonstration. L'inégalité $p_{1,3} \leq \min\{p_1, p_3\}$ vient du fait que les protocoles \mathcal{M}_1 et \mathcal{M}_3 ne peuvent pas avoir un taux d'activité commune supérieure au taux d'activité du plus petit des deux. L'inégalité $p_{1,3} \geq \max\{0, p_{1,2} + p_{2,3} - 1\}$ vient du fait que dans le pire des cas, l'essentiel des activités de \mathcal{M}_1 et de \mathcal{M}_3 sont disjointes. Si $p_{1,2} + p_{2,3}$ dépasse 1, c'est que \mathcal{M}_1 et \mathcal{M}_3 partagent obligatoirement des activités communes. \square

Le tableau 4.1 donne l'intervalle des valeurs de $p_{1,3}$ et les valeurs de $p_{1,2} \cdot p_{2,3}$, pour quelques valeurs particulières¹³. On peut remarquer que, dans certains cas, l'intervalle des valeurs de $p_{1,3}$ possède des valeurs supérieures à $p_{1,2} \cdot p_{2,3}$: cela signifie que le passage par \mathcal{M}_2 peut être avantageux dans certains cas. Ces probabilités n'ont pas à être connues des nœuds, mais peuvent servir lors de la conception des protocoles (et de l'automate des nœuds) pour évaluer la probabilité de rendez-vous des nœuds, et éventuellement faciliter la transition en concevant un protocole MAC intermédiaire ne servant qu'à la transition.

$p_{1,2}$	$p_{2,3}$	p_2	$p_{1,3}$	$p_{1,2} \cdot p_{2,3}$
10%	40%	100%	[0% ; 10%]	4%
10%	40%	70%	[0% ; 40%]	4%
10%	40%	40%	[0% ; 70%]	4%
60%	80%	100%	[40% ; 60%]	48%
60%	80%	80%	[40% ; 80%]	48%
60%	80%	60%	[40% ; 100%]	48%

Tableau 4.1 – Comparaison des probabilités $p_{1,3}$ et $p_{1,2} \cdot p_{2,3}$ pour des valeurs particulières.

¹³. On fait ici l'hypothèse que \mathcal{M}_1 , \mathcal{M}_2 et \mathcal{M}_3 sont indépendants.

4.2.3.3 Protocoles réseaux qui supportent le changement d'état

Nous étudions à présent l'impact des changements d'état sur les protocoles réseaux. Nous cherchons à proposer un mécanisme de changement d'état qui évite les boucles de routage.

Supposons que les nœuds du réseau sont en train de passer d'un état 1 (utilisant le protocole de routage \mathcal{R}_1) à un état 2 (utilisant le protocole de routage \mathcal{R}_2). Chaque nœud est donc soit dans l'état 1, soit dans l'état 2. Nous supposons que chaque nœud connaît l'état de ses voisins (cette information pouvant être transmise par les protocoles de routage). Un paquet donné est routé par \mathcal{R}_1 ou par \mathcal{R}_2 , selon les nœuds rencontrés sur le chemin.

Lorsqu'un nœud n achemine un paquet pour une destination d , il peut se produire quatre cas en fonction de l'état du nœud n et de l'état du nœud correspondant au prochain saut pour d :

- n est dans l'état 1 et $\mathcal{R}_1(n, d)$ est dans l'état 1. n et $\mathcal{R}_1(n, d)$ n'ont pas encore été prévenus du changement d'état, mais ils sont tous les deux dans un état cohérent : n peut acheminer le paquet.
- n est dans l'état 2 et $\mathcal{R}_2(n, d)$ est dans l'état 2. n et $\mathcal{R}_2(n, d)$ sont dans l'état attendu : n peut (et doit) acheminer le paquet.
- n est dans l'état 1 et $\mathcal{R}_1(n, d)$ est dans l'état 2. Les états des nœuds ne sont pas cohérents, mais le prochain saut est dans l'état attendu. n peut envoyer le paquet à $\mathcal{R}_1(n, d)$ ou attendre de passer dans l'état 2.
- n est dans l'état 2 et $\mathcal{R}_2(n, d)$ est dans l'état 1. Les états des nœuds ne sont pas cohérents, mais c'est n qui est dans l'état attendu. À terme, $\mathcal{R}_2(n, d)$ passera aussi dans cet état 2. n peut attendre que $\mathcal{R}_2(n, d)$ passe dans l'état 2, ou envoyer le paquet à $\mathcal{R}_2(n, d)$ mais au risque de causer une boucle (si les nœuds dans l'état 1 envoient à des prochains sauts dans l'état 2), voire d'empêcher à la destination d de recevoir le paquet si la boucle est infinie.

On peut remarquer que les boucles n'apparaissent que lorsque la frontière entre les nœuds dans l'état 1 et les nœuds dans l'état 2 est franchie plusieurs fois. En interdisant à un paquet de franchir la frontière de l'état 2 vers l'état 1, les boucles peuvent être évitées.

Notons aussi que lorsque les protocoles de routage \mathcal{R}_1 et \mathcal{R}_2 sont compatibles, aucune boucle ne peut se produire (même si la frontière entre les nœuds dans l'état 1 et dans l'état 2 est franchie plusieurs fois).

Définition 15 (Protocole de routage interne). Notons N l'ensemble des nœuds pour lequel le protocole de routage \mathcal{R} est défini. \mathcal{R} est dit interne si, pour toute destination $d \in N$ et pour tout nœud $n \in N$, $\mathcal{R}(n, d) \in N$.

Lorsque le protocole de routage \mathcal{R}_2 est interne et que la destination d est dans l'état 2, un nœud de l'état 2 ne peut pas envoyer une trame à un nœud dans l'état 1. Les boucles sont alors impossibles.

Généralement, les protocoles de routage sont internes. Il existe toutefois des contre-exemples : lorsqu'une entité centralisée met à jour les tables de routage de tous les nœuds sans suivre un ordre topologique à partir de la destination d , on peut avoir une situation où le prochain saut selon la nouvelle table n'a pas encore eu sa propre table mise à jour.

Pour éviter les boucles lorsque les protocoles de routage ne sont pas compatibles et que le protocole de routage \mathcal{R}_2 n'est pas interne, nous proposons l'algorithme 2. Dans cet algorithme, on peut remarquer que lorsqu'un nœud n est dans l'état 1, il envoie toujours à $\mathcal{R}_1(n, d)$, quelque soit l'état de $\mathcal{R}_1(n, d)$. Cet algorithme ne conduit à aucune boucle : plus précisément, cet algorithme rend les protocoles \mathcal{R}_1 et \mathcal{R}_2 retardables. Notons que cet algorithme ne fait pas l'hypothèse que $\mathcal{R}_1(n, d)$ et $\mathcal{R}_2(n, d)$ soient joignables immédiatement par n . Cette propriété est importante, car il est possible que n et $\mathcal{R}_1(n, d)$ (ou n et $\mathcal{R}_2(n, d)$) ne soient pas dans le même état, et utilisent des calendriers d'activités incompatibles¹⁴.

Algorithme 2 : Protocole de routage supportant le changement d'état.

Entrées : p est un paquet à envoyer, n est le nœud en cours
 $d \leftarrow$ destination du paquet p
si n est dans l'état 1 **alors**
 | n achemine la trame via $\mathcal{R}_1(n, d)$
sinon
 | **si** $\mathcal{R}_2(n, d)$ est dans l'état 1 **alors**
 | n conserve la trame tant que $\mathcal{R}_2(n, d)$ est dans l'état 1
 | **sinon**
 | n achemine la trame via $\mathcal{R}_2(n, d)$
 | **finsi**
finsi

L'algorithme 2 peut être généralisé en autorisant le paquet à traverser la frontière de \mathcal{R}_2 à \mathcal{R}_1 au plus x fois, x étant un paramètre du protocole. Lorsque $x \geq 1$, des boucles temporaires peuvent apparaître, mais un même nœud ne peut pas recevoir le même paquet plus de x fois. Cette modification nécessite d'inclure la valeur du nombre de traversées de frontières restantes dans chaque paquet.

4.2.4 Bilan des architectures globales

Les architectures multi-piles globales sont des architectures dans lesquelles la pile protocolaire utilisée par les nœuds dépend des événements détectés dans le réseau plutôt que de critères locaux aux nœuds. Nous avons décrit dans cette partie une architecture globale liée au projet Clervolc, et plus particulièrement à un ensemble de protocoles visant à réaliser un compromis entre une grande économie d'énergie quand aucun événement significatif n'est détecté, et une grande réactivité quand les événements détectés sont critiques.

Dans le futur, de nombreuses architectures multi-piles globales peuvent être envisagées, en modifiant les hypothèses qui ont été faites. Par exemple, il peut être intéressant d'étudier le comportement du réseau quand deux événements contradictoires sont détectés par des nœuds différents (une prise en compte de ce phénomène

14. Les cas où le nœud et le prochain saut ne peuvent pas communiquer sont temporaires : dans le cas où cette situation est due à une incohérence entre l'état du nœud et l'état de son prochain saut, cette incohérence finit par disparaître lorsque le nœud dans l'état 1 passe dans l'état 2.

au niveau MAC est décrite rapidement dans la partie 4.1.3.2). Il est probable qu’Af-foua Thérèse ABY étudie cet aspect au cours de sa thèse. Une autre perspective est d’étudier le comportement du réseau quand l’état des nœuds peut évoluer rapidement, et plus précisément quand plus de deux états différents peuvent coexister simultanément dans le réseau.

4.3 Étude de l’évolution du réseau

Dans une architecture multi-piles globale, l’état du réseau évolue en fonction du changement d’état de chacun des nœuds. En effet, lorsqu’un évènement nécessitant un changement d’état se produit (il peut s’agir d’un évènement significatif qui apparaît dans l’environnement et qui est détecté par un nœud capteur, ou d’un message de reconfiguration émanant d’un opérateur), les nœuds changent d’état un à un, jusqu’à ce que tous les nœuds atteignent l’état attendu.

Dans cette partie, nous étudions la manière dont le réseau évolue en modélisant cette évolution comme un graphe (chaque nœud du graphe modélisant un état du réseau) [GR13*] : ces résultats sont préliminaires, mais ouvrent des perspectives intéressantes de recherche. Dans la partie 4.3.1, nous étudions les propriétés du graphe représentant l’évolution du réseau, et montrons qu’il s’agit d’un treillis. Dans la partie 4.3.2, nous donnons des algorithmes permettant de manipuler ce treillis. Ces algorithmes permettent par exemple de calculer le temps maximum pendant lequel il existe des nœuds dans plusieurs états (ce qui permet de connaître la durée minimale ou maximale de transition entre deux états du réseau, et ainsi donner un temps minimum de changement d’état dans l’automate des états d’un nœud), ou d’identifier les états possibles du réseau à partir d’une connaissance partielle (ce qui peut être utile pour les applications utilisant le réseau de capteurs sans fil).

4.3.1 Étude du graphe d’évolution

L’évolution du réseau se fait en fonction du changement d’état des nœuds, initié par des nœuds particuliers que nous appelons graines. Un nœud n ne peut changer d’état que dans les cas suivants :

- n est une graine et détecte un changement d’état (suite à la détection d’un évènement significatif, ou à la réception d’une commande d’un opérateur),
- n est un voisin d’un nœud ayant changé d’état (ce qui permet de modéliser les échanges réseaux).

Nous considérons initialement que le temps pris par un nœud pour changer d’état (qu’il soit une graine ou le voisin d’un nœud ayant changé d’état) est fini mais arbitraire. Dans le cas d’une graine, le temps pris dépend de l’évènement surveillé. Dans le cas d’un voisin d’un nœud ayant changé d’état, le temps pris dépend du protocole de mise à jour de l’état, qui peut être géré par la couche application du nœud ou par des couches inférieures (comme la couche réseau), puis des contraintes temporelles induites par la sous-couche MAC.

Soit G le graphe représentant le réseau de capteurs sans fil. Nous appelons état de G l’affectation d’un état à chacun des nœuds de G . Le graphe d’évolution de G

à partir d'une graine a , noté $\mathcal{T}(G, a)$, est le graphe dans lequel chaque nœud est un état possible de G , et dans lequel un arc relie un nœud E_1 (de $\mathcal{T}(G, a)$) à un nœud E_2 (de $\mathcal{T}(G, a)$) si et seulement si le réseau G peut passer de l'état E_1 à l'état E_2 directement. Nous noterons les sommets de G par des lettres minuscules, les ensembles de sommets (ce qui inclue les états de $\mathcal{T}(G, a)$) par des lettres capitales, et les ensembles d'ensembles de sommets par des lettres capitales et une police de caractères imitant l'écriture calligraphiée.

Par exemple, considérons le graphe G de quatre nœuds représenté sur la figure 4.12. Le nœud a est supposé être l'origine de l'évènement (à venir) nécessitant un changement de état (a peut être un nœud capteur ou le puits applicatif). La figure 4.13 représente le graphe d'évolution $\mathcal{T}(G, a)$, qui correspond aux évolutions possibles de l'état des nœuds de G , les nœuds de l'état 1 étant représentés en blanc et ceux de l'état 2 en gris. Initialement, tous les nœuds sont dans l'état 1. À un instant donné, le nœud a détecte un évènement significatif ou reçoit une commande de l'opérateur : a passe dans l'état 2. Puis, a informe l'un de ses voisins (b ou d) de la nécessité de changer d'état. Quand a et b sont tous les deux dans l'état 2, c ou d peuvent être informés. En revanche, quand a et d sont tous les deux dans l'état 2, seul b peut être informé (que ce soit par a ou par d). À terme, tous les nœuds du réseau G atteignent l'état 2.

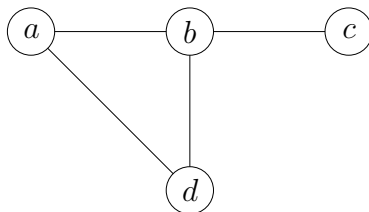


Figure 4.12 – Un graphe G représentant un réseau de capteurs sans fil.

Pour rendre la représentation des graphes d'évolution plus compacte, nous représentons dans la suite de ce document uniquement la liste des nœuds étant dans l'état 2 dans chaque nœud de $\mathcal{T}(G, a)$ ¹⁵, en concaténant le nom des nœuds et en les représentant entre parenthèses. La figure 4.14 représente le graphe $\mathcal{T}(G, a)$ avec cette nouvelle notation.

Dans la partie 4.3.1.1, nous étudions la structure du graphe d'évolution sous des hypothèses variées, et nous montrons qu'il s'agit d'un treillis. Dans la partie 4.3.1.2, nous étudions les propriétés de certains nœuds de ce treillis.

4.3.1.1 Structure du graphe d'évolution

Dans la suite, nous étudions la structure du graphe d'évolution en considérant plusieurs jeux d'hypothèses. Le premier jeu d'hypothèse représente notre cas de base. Les autres jeux d'hypothèses servent uniquement à la discussion¹⁶.

15. Cette représentation est légèrement modifiée lorsque les nœuds peuvent être dans trois états ou plus.

16. Certains jeux d'hypothèses sont étudiés dans l'annexe A.1.

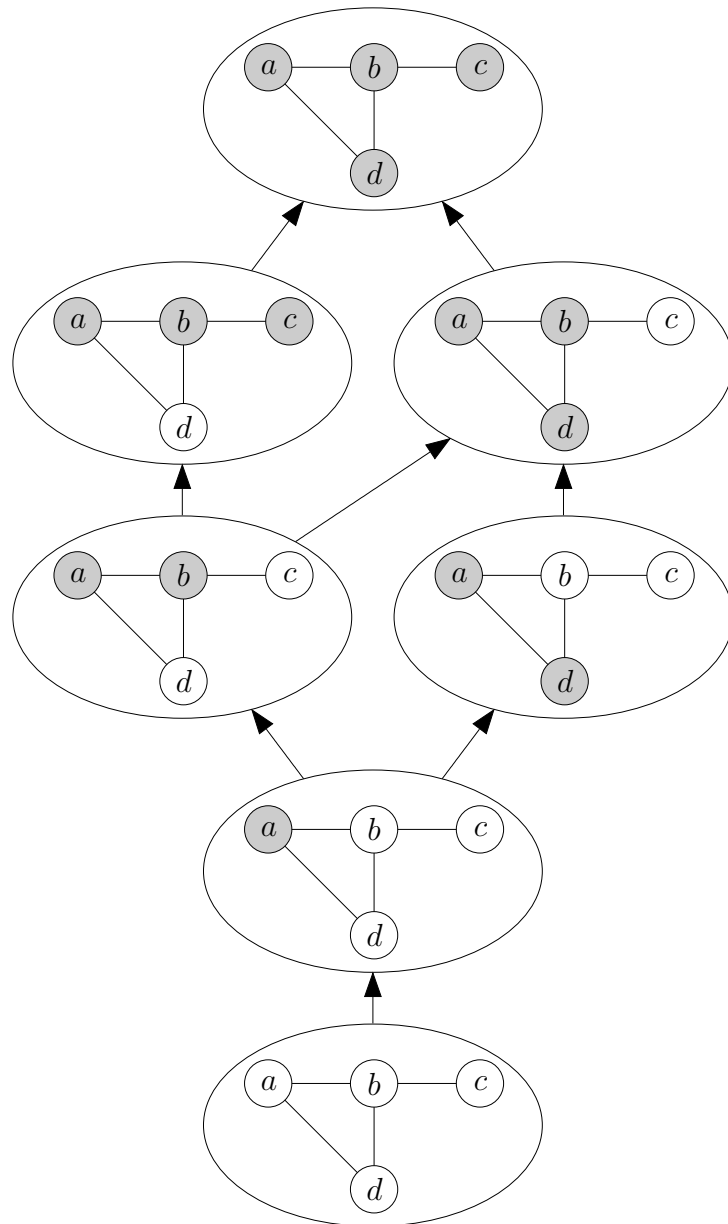


Figure 4.13 – Graphe d'évolution $\mathcal{T}(G, a)$ pour le graphe G représenté sur la figure 4.12, quand a est l'initiateur du changement d'état. Les nœuds en blanc sont dans l'état 1, et les nœuds en gris sont dans l'état 2.

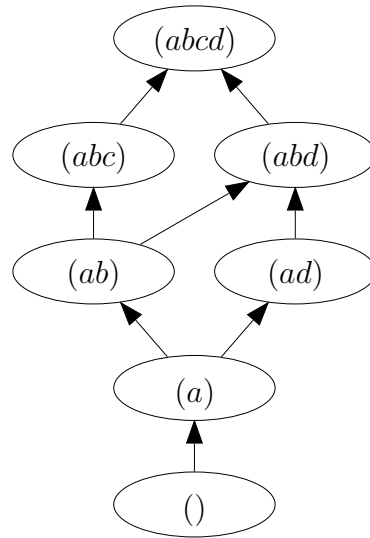


Figure 4.14 – Graphe d'évolution $\mathcal{T}(G, a)$ pour le graphe G représenté sur la figure 4.12.

Évolution unitaire mono-graine et bi-états dans un graphe connexe.

Définition 16 (Évolution unitaire). L'évolution unitaire correspond au cas où un seul nœud à la fois est autorisé à changer d'état. En d'autres termes, l'évolution unitaire fait l'hypothèse que la probabilité que deux changements d'état aient lieu simultanément est nulle¹⁷.

Définition 17 (Évolution mono-graine). L'évolution mono-graine correspond au cas où un seul nœud (connu *a priori*) est la graine.

Définition 18 (Évolution bi-états). L'évolution bi-états correspond à la situation où chaque nœud est soit dans l'état 1, soit dans l'état 2. Tous les nœuds sont initialement dans l'état 1, et un nœud qui passe dans l'état 2 ne peut pas revenir à l'état 1. L'automate des nœuds est donc linéaire à deux états¹⁸.

Rappelons qu'un treillis est un ensemble partiellement ordonné dans lequel chaque couple d'éléments a une borne supérieure et une borne inférieure [Bir49].

Théorème 4. Soit G un graphe connexe et a une graine. Quand l'évolution est unitaire, mono-graine et bi-états, le graphe d'évolution $\mathcal{T}(G, a)$ forme un treillis.

Démonstration. Montrons tout d'abord que la famille d'ensembles formée des états de $\mathcal{T}(G, a)$ est fermée par union. Soit X et Y deux états de $\mathcal{T}(G, a)$. Par construction, $\mathcal{T}(G, a)$ est connexe : il existe donc un chemin \mathcal{P}_1 dans $\mathcal{T}(G, a)$ qui relie \emptyset à X , et un chemin \mathcal{P}_2 qui relie \emptyset à Y , comme schématisé sur la figure 4.15 (représentant une vision partielle de $\mathcal{T}(G, a)$). Notons k la longueur de \mathcal{P}_2 , et construisons les états Z_i , pour i de 0 à k , de la manière suivante :

17. La généralisation de l'évolution unitaire est l'évolution multiple. Elle est détaillée dans l'annexe A.1.

18. La généralisation de l'évolution bi-états est l'évolution multi-états. Elle est détaillée dans l'annexe A.1.

- $Z_0 = X$,
- $Z_{i+1} = Z_i \cup \{n_i\}$, avec n_i est le i -ème nœud du chemin \mathcal{P}_2 . Notons ici que si $n_i \in Z_i$, alors $Z_{i+1} = Z_i$.

On a bien $Z_k = X \cup Y$. Il reste à montrer que l'état Z_k est un état de $\mathcal{T}(G, a)$. Nous allons le montrer par induction, en montrant que si $Z_i \in \mathcal{T}(G, a)$, alors $Z_{i+1} \in \mathcal{T}(G, a)$ (et en notant que $Z_0 = X \in \mathcal{T}(G, a)$). Supposons que $Z_i \in \mathcal{T}(G, a)$ pour un certain i . Si $n_i \in Z_i$, $Z_{i+1} \in \mathcal{T}(G, a)$ de manière triviale. Sinon, $n_i \notin Z_i$, et il existe un nœud x de Z_i qui est voisin de n_i (car ce nœud x existe pour informer n_i dans \mathcal{P}_2), ce qui signifie que l'arc de Z_i à Z_{i+1} existe bien dans $\mathcal{T}(G, a)$.

Comme $\mathcal{T}(G, a)$ contient l'ensemble vide (qui est l'état initial) et est fermé par union, il s'agit d'une co-famille de Moore, et $\mathcal{T}(G, a)$ est donc un treillis [Bir67]¹⁹. \square

Cette propriété a été prouvée préalablement dans le lemme 3.4 de [KS96], dans lequel les auteurs démontrent que si G est un graphe connexe, pour tout $X \in \mathcal{C}(G)$ tel que $X \neq \emptyset$, le filtre principal généré par X est un treillis complet semi-modulaire. Dans ce lemme, $\mathcal{C}(G)$ correspond à l'ensemble de tous les sous-graphes connexes de G , ordonnés par inclusion. Le filtre principal généré par X , noté $X\uparrow$ par les auteurs, est égal à $\{E \in \mathcal{C}(G) \mid X \leq E\}$, c'est-à-dire à l'ensemble des sous-graphes connexes de G contenant au moins les nœuds de l'ensemble X . Dans le cas où $X = \{a\}$, on a $X\uparrow = \mathcal{T}(G, a)$. Le résultat du lemme 3.4 de [KS96] est donc compatible avec le nôtre²⁰.

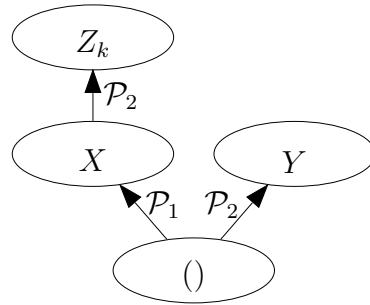


Figure 4.15 – Schéma représentant quelques états du graphe d'évolution, et permettant de montrer que l'ensemble des états est fermé par union.

Dans la suite, si A et B sont deux états du treillis, nous notons $A \leq B$ si $A \subset B$, $A \wedge B$ le supremum de A et B (c'est-à-dire, la plus petite borne supérieure de A et B), et $A \vee B$ l'infimum de A et B (c'est-à-dire, la plus grande borne inférieure de A et B). Nous avons $A \vee B = A \cup B$. En revanche, $A \wedge B \subset A \cap B$. L'élément maximum du treillis est noté \top , et l'élément minimum est noté \perp .

19. La preuve de [Bir67] s'applique en fait aux familles de Moore, mais on peut passer d'une famille de Moore à une co-famille de Moore par complémentation, les deux ensembles de familles étant en bijection.

20. Ce lemme indique de plus que le treillis $\mathcal{T}(G, a)$ est complet et semi-modulaire, ce qui implique qu'il est gradué. Ces trois propriétés (complétude, semi-modularité et graduation) sont démontrées dans l'annexe A.2, en utilisant une approche différente, basée sur les liens entre la topologie G et le treillis $\mathcal{T}(G, a)$.

Évolution dans un graphe non connexe.

Lorsque le graphe n'est pas connexe, le résultat précédent s'obtient en remplaçant G par la composante connexe de G qui contient la graine a .

Évolution multi-graines.

Définition 19 (Évolution multi-graines). L'évolution multi-graines correspond au cas où plusieurs nœuds (connus *a priori*) sont des graines. Chaque graine peut changer d'état à un instant arbitraire.

Propriété 14. Soit G un graphe connexe et A un ensemble de graines. Quand l'évolution est unitaire, multi-graines et bi-états, le graphe d'évolution $\mathcal{T}(G, A)$ forme un treillis.

Démonstration courte. La démonstration de cette propriété est similaire à celle du théorème 4. En fait, il existe une relation forte entre le graphe d'évolution à partir d'une seule graine et le graphe d'évolution à partir d'un ensemble A de graines. Pour s'en convaincre, construisons le graphe G' à partir du graphe G , en créant un nœud virtuel a dans G' et une arête de a à chaque graine de A . D'après le théorème 4, le graphe $\mathcal{T}(G', a)$ est un treillis. Le graphe $\mathcal{T}(G, A)$ est très proche de $\mathcal{T}(G', a)$: $\mathcal{T}(G, A)$ est égal à $\mathcal{T}(G', a)$ sans l'état \emptyset , et dans lequel chaque occurrence du nœud virtuel a a été supprimée. Une illustration est donnée sur l'exemple du graphe G de la figure 4.16 (la partie en pointillés n'appartenant qu'au graphe G'), avec $\mathcal{T}(G, A)$ et $\mathcal{T}(G', a)$ représentés sur la figure 4.17. \square

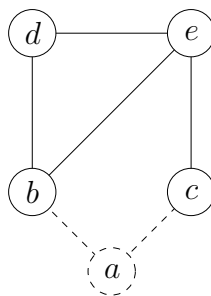


Figure 4.16 – Un exemple de graphe G avec quatre nœuds, et de graphe G' avec le nœud virtuel a et des arêtes supplémentaires de a aux graines b et c .

Ce résultat est différent de celui du lemme 3.4 de [KS96], avec $X = A$. En effet, quand $X = A$, X^\uparrow est le treillis dans lequel chaque état contient toutes les graines de A , alors que les états de $\mathcal{T}(G, A)$ contiennent tous au moins une graine (à l'exception de l'état \emptyset). Ce résultat est aussi différent de celui donné dans [Ste90], où l'auteur indique que pour un graphe G , $\mathcal{C}(G)$ (défini comme dans [KS96]) est un treillis si et seulement si G est connexe et chaque cycle de G induit un sous-graphe complet de G . La preuve de [Ste90] nécessite des conditions sur les cycles de G alors que nous ne le nécessitons pas, car $\mathcal{C}(G)$ et $\mathcal{T}(G, V)$ (avec V représentant l'ensemble

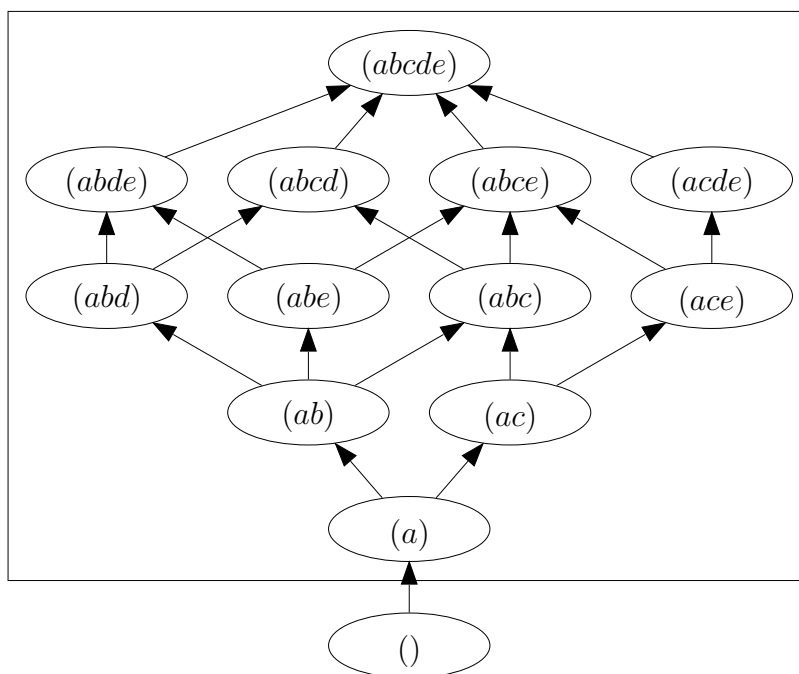


Figure 4.17 – Le treillis $\mathcal{T}(G', a)$, pour le graphe G' de la figure 4.16 et la graine virtuelle a , est très proche du treillis $\mathcal{T}(G, A)$ (encadré, sans occurrence de a), pour le graphe G de la figure 4.16 et l'ensemble de graines $A = \{b, c\}$.

des nœuds de G), ne sont pas construits de la même manière²¹. En effet, dans sa preuve, l'auteur indique que si a et b sont deux nœuds de G qui sont sur un cycle, il existe deux ensembles de nœuds S_1 et S_2 qui couvrent a et b et dont l'intersection est $\{a, b\}$. L'état $\{a, b\}$ doit donc appartenir à $\mathcal{C}(G)$ (puisque $\mathcal{C}(G)$ est un treillis), ce qui nécessite que a et b soient reliés par une arête dans G . Dans notre cas, l'état $\{a, b\}$ appartient bien à $\mathcal{T}(G, V)$, mais le fait que a et b soient deux graines suffit à prouver l'existence de cet état. Notre construction de $\mathcal{T}(G, V)$ n'impose donc pas que a et b soient reliés par une arête.

Quelques propriétés supplémentaires du graphe d'évolution sont étudiées dans l'annexe A.2.

4.3.1.2 Propriétés des nœuds du graphe d'évolution

Dans cette partie, nous caractérisons certains états du graphe d'évolution, en identifiant les états sup-irréductibles et inf-irréductibles de $\mathcal{T}(G, a)$ en fonction de la topologie du réseau G (et de la graine a). Nous faisons le rapprochement entre ces états particuliers de $\mathcal{T}(G, a)$ et les propriétés de la topologie G . Certaines propriétés supplémentaires sont étudiées dans l'annexe A.3.

États sup-irréductibles.

Un état sup-irréductible de $\mathcal{T}(G, a)$ est un état qui n'a qu'un seul prédécesseur.

21. D'ailleurs, $\mathcal{C}(G)$ et $\mathcal{T}(G, V)$ sont très différents : les états de $\mathcal{T}(G, V)$ sont l'ensemble des parties de V , alors que $\mathcal{C}(G)$ ne contient que les sous-graphes connexes de G .

Propriété 15. Soit G un graphe, a la graine et $\mathcal{T}(G, a)$ le treillis correspondant. Les deux propositions suivantes sont équivalentes :

- E est un état sup-irréductible,
- $G(E)$ est un chemin de G qui contient a comme extrémité, avec $G(E)$ le sous-graphe de G induit²² par les nœuds de E .

Démonstration. Montrons que si $G(E)$ est un chemin de G qui contient a comme extrémité, alors E est un sup-irréductible du treillis. Notons $P = (x_0 = a, x_1, \dots, x_k)$ le chemin $G(E)$. L'état $E = \{x_0, x_1, \dots, x_k\}$ est bien un état du treillis, et l'état $\{x_0, x_1, \dots, x_{k-1}\}$ est bien son prédécesseur. Nous allons montrer que E n'a pas d'autre prédécesseur en raisonnant par l'absurde. Supposons qu'il existe un autre état E' , prédécesseur de E . Alors, il existe y tel que $E' \cup \{y\} = E$ (avec $y \in \{x_0, \dots, x_{k-1}\}$). Il existe donc un indice $i \in [1; k-1]$ tel que $y = x_i$ (car $a = x_0 \in E'$ par construction). On a $E' = E \setminus \{x_i\}$. Considérons le sous-graphe $G(E')$: ce graphe est égal au sous-graphe $G(E)$ auquel on supprime le nœud x_i . Or, $G(E)$ était un chemin avec $a = x_0$ et x_k comme extrémités, et on lui a supprimé un nœud intermédiaire x_i avec $i \in [1; k-1]$. Ainsi, $G(E')$ n'est pas un graphe connexe, et E' ne peut donc pas être un état du treillis. On aboutit à une contradiction : E a un seul prédécesseur, ce qui signifie que E est un sup-irréductible du treillis.

Montrons à présent que si E est un sup-irréductible du treillis, $G(E)$ est un chemin de G qui contient a comme extrémité. Soit E un sup-irréductible du treillis. Comme E est un état du treillis, $G(E)$ est un sous-graphe connexe.

- Nous allons montrer par l'absurde que $G(E)$ ne contient pas de cycle de longueur trois ou plus. Supposons que $G(E)$ contienne un cycle (y_0, y_1, \dots, y_m) avec $m \geq 3$. Puisque $m \geq 3$, il existe au moins deux nœuds y' et y'' tels que $y' \neq a$ et $y'' \neq a$. Comme ces nœuds appartiennent à un cycle, $G(E \setminus \{y'\})$ est un graphe connexe qui contient a , et $G(E \setminus \{y''\})$ aussi. Il existe donc deux états différents, $E' = E \setminus \{y'\}$ et $E'' = E \setminus \{y''\}$, qui sont des prédécesseurs immédiats de l'état E . Donc, E n'est pas un sup-irréductible, ce qui est une contradiction. $G(E)$ ne contient pas de cycle de longueur trois ou plus.
- $G(E)$ ne contient pas de cycle de longueur un, étant donné que G ne contient aucune arête du type (x, x) .
- $G(E)$ ne contient pas de cycle de longueur deux, car G ne contient au plus qu'une seule arête du type (x, y) , x et y étant deux nœuds quelconques de G .
- Nous allons montrer que $G(E)$ est un chemin. Nous savons que $G(E)$ est un arbre (car il s'agit d'un graphe connexe sans cycle). Raisonnons par l'absurde, et faisons l'hypothèse que $G(E)$, enraciné en a , contient (au moins) deux branches en un certain nœud. Soit y' le nœud terminal de l'une de ces branches, et y'' le nœud terminal de l'autre branche (en notant que $y' \neq a$ et $y'' \neq a$). Il existe deux états différents, $E' = E \setminus \{y'\}$ et $E'' = E \setminus \{y''\}$, qui sont des prédécesseurs immédiats de l'état E . Donc, E n'est pas un sup-irréductible, ce qui est une contradiction. $G(E)$ est un arbre qui ne contient qu'une seule branche, c'est-à-dire un chemin.

□

22. Rappelons que le graphe induit par un ensemble de nœuds X est le sous-graphe de G dont X est l'ensemble de nœuds et dont les arêtes ont leurs extrémités dans X .

États inf-irréductibles.

Un état inf-irréductible de $\mathcal{T}(G, a)$ est un état qui n'a qu'un seul successeur.

Propriété 16. *Soit G un graphe et a une graine. Soit $x \neq a$ un nœud séparateur de G , c'est-à-dire tel que la suppression de x dans G sépare G en (au moins) deux graphes connexes. Notons G' la composante connexe de $G \setminus \{x\}$ qui contient a . Alors, l'état S' est un inf-irréductible de $\mathcal{T}(G, a)$, où S' représente l'ensemble des nœuds de G' .*

Démonstration. Soit G un graphe, a une graine, $x \neq a$ un nœud séparateur de G et G' la composante connexe de $G \setminus \{x\}$ qui contient a .

G' est une composante connexe du graphe qui contient a . Le treillis $\mathcal{T}(G, a)$ contient donc l'état $E' = S'$, constitué des nœuds de G' . x est connecté à l'un des nœuds de S' (au moins), étant donné que x est séparateur de G . Ainsi, l'état $E = S' \cup \{x\}$ est bien un successeur de l'état E' . Montrons qu'il est le seul.

Supposons qu'il existe un successeur E'' à E' , différent de E . Il existe donc un nœud $y \neq x$ tel que $E'' = S' \cup \{y\}$. y appartient à une composante connexe de G qui n'est pas G' . Ceci n'est possible que si $y = x$, ce qui est impossible. Ainsi, E' n'a qu'un seul successeur : E' est donc un inf-irréductible. \square

Propriété 17. *Soit G un graphe, a une graine, et $\mathcal{T}(G, a)$ le treillis correspondant. Si E est un état inf-irréductible de $\mathcal{T}(G, a)$, alors soit il s'agit d'un co-atome, soit le nœud $\text{Succ}(E) \setminus E$ est un séparateur de G (avec $\text{Succ}(E)$ qui désigne l'unique successeur immédiat de E).*

Démonstration. Soit E un état inf-irréductible de $\mathcal{T}(G, a)$ qui n'est ni un co-atome, ni tel que le sommet $x = \text{Succ}(E) \setminus E$ est un séparateur de G . On sait que l'état $\text{Succ}(E) = E \cup \{x\}$ existe. Montrons qu'il existe un autre successeur à E . $V \setminus E$ n'est pas réduit à x , avec V représentant l'ensemble des nœuds de G , sinon E serait un co-atome de $\mathcal{T}(G, a)$. Il existe donc des nœuds dans $(G \setminus E) \setminus \{x\}$. Comme x n'est pas un séparateur de G , il existe des arêtes entre le graphe représenté par E (qui est connexe par construction de E), et le graphe $(G \setminus E) \setminus \{x\}$. Il existe donc $y_1 \in E$ et $y_2 \in V \setminus E \setminus \{x\}$ tels que y_1 et y_2 sont reliés par une arête. Alors, $E \cup \{y_2\}$ est un successeur immédiat de E , avec $E \cup \{y_2\} \neq E \cup \{x\}$. Ainsi, E n'est pas un inf-irréductible, ce qui est en contradiction avec l'hypothèse. \square

Propriété 18. *Soit a une graine et $x \neq a$ un nœud de G . x n'est pas un séparateur de G est équivalent à $V \setminus \{x\}$ est un co-atome de $\mathcal{T}(G, a)$, avec V l'ensemble des nœuds de G .*

Démonstration. Montrons tout d'abord que si x n'est pas un séparateur de G , alors $V \setminus \{x\}$ est un co-atome de $\mathcal{T}(G, a)$. Comme x n'est pas un séparateur de G , le graphe $G \setminus \{x\}$ est connexe (et contient la graine a). L'état $V \setminus \{x\}$ existe donc dans $\mathcal{T}(G, a)$. Le successeur de cet état est V . $V \setminus \{x\}$ est bien un co-atome de $\mathcal{T}(G, a)$.

Montrons à présent que si $V \setminus \{x\}$ est un co-atome de $\mathcal{T}(G, a)$, alors x n'est pas un séparateur de G . Comme $V \setminus \{x\}$ est un état de $\mathcal{T}(G, a)$, c'est que tous les nœuds de $V \setminus \{x\}$ ont été informés à partir de a , ce qui signifie que $V \setminus \{x\}$ est connexe. Donc, x n'est pas un séparateur de G . \square

4.3.2 Algorithmes concernant l'évolution

Dans cette partie, nous proposons plusieurs algorithmes permettant de manipuler le graphe d'évolution $\mathcal{T}(G, a)$ d'un réseau G . Nous proposons tout d'abord des algorithmes de construction de ce graphe d'évolution. Dans la partie 4.3.2.1, nous montrons comment construire $\mathcal{T}(G, a)$ à partir du graphe G et de la graine a . Dans la partie 4.3.2.2, nous décrivons des algorithmes permettant de mettre à jour $\mathcal{T}(G, a)$ suite à des mises à jour de G . Nous proposons ensuite des algorithmes permettant de faire des calculs sur le graphe d'évolution. Dans la partie 4.3.2.3, nous calculons le temps maximum pour que tous les nœuds du réseau passent de l'état 1 à l'état 2. Dans la partie 4.3.2.4, nous calculons les états possibles dans lequel le réseau peut être à partir de la connaissance de l'état de certains nœuds.

4.3.2.1 Construction du treillis

L'algorithme 3 permet de construire le treillis $\mathcal{T}(G, a)$ à partir du graphe G et de la graine a . L'algorithme procède en utilisant une liste d'états non traités, notée \mathcal{F} . Cette liste est initialisée avec l'état \emptyset , où tous les nœuds sont dans l'état 1. L'état successeur de l'état \emptyset est l'état où seule la graine est dans l'état 2. Ensuite, chaque état E a pour successeurs les états E' tels que $E' = E \cup \{v\}$, avec v un voisin d'un des nœuds de E , n'appartenant pas déjà à E .

Algorithme 3 : Algorithme de construction de $\mathcal{T}(G, a)$.

Entrées : G est un graphe, $a \in G$ est la graine
 créer l'état \emptyset dans $\mathcal{T}(G, a)$
 créer l'état $\{a\}$ dans $\mathcal{T}(G, a)$
 ajouter $\{a\}$ comme successeur de \emptyset dans $\mathcal{T}(G, a)$
 $\mathcal{F} \leftarrow \{\{a\}\}$
tantque $\mathcal{F} \neq \emptyset$ **faire**
 $E \leftarrow$ un état de \mathcal{F}
 pour chaque $n \in E$ **faire**
 pour chaque voisin v de n dans G **faire**
 $E' \leftarrow E \cup \{v\}$
 si $E' \neq E$ **alors**
 si E' n'existe pas dans $\mathcal{T}(G, a)$ **alors**
 créer l'état E' dans $\mathcal{T}(G, a)$
 finsi
 ajouter E' comme successeur de E dans $\mathcal{T}(G, a)$
 finsi
 finpourchaque
 finpourchaque
 $\mathcal{F} \leftarrow \mathcal{F} \setminus \{E\}$
fintantque
retourner $\mathcal{T}(G, a)$

4.3.2.2 Mise à jour du treillis suite à une mise à jour du graphe

Nous allons à présent donner des algorithmes permettant de mettre à jour le treillis $\mathcal{T}(G, a)$ en fonction des mises à jour du graphe G . À titre d'exemple, nous partirons du graphe G représenté sur la figure 4.18, correspondant au treillis représenté sur la figure 4.19.

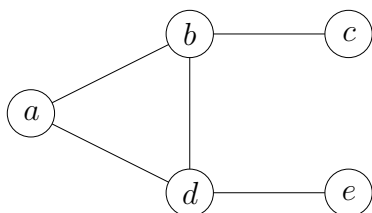


Figure 4.18 – Graphe G sur lequel nous allons illustrer les algorithmes de mise à jour du treillis.

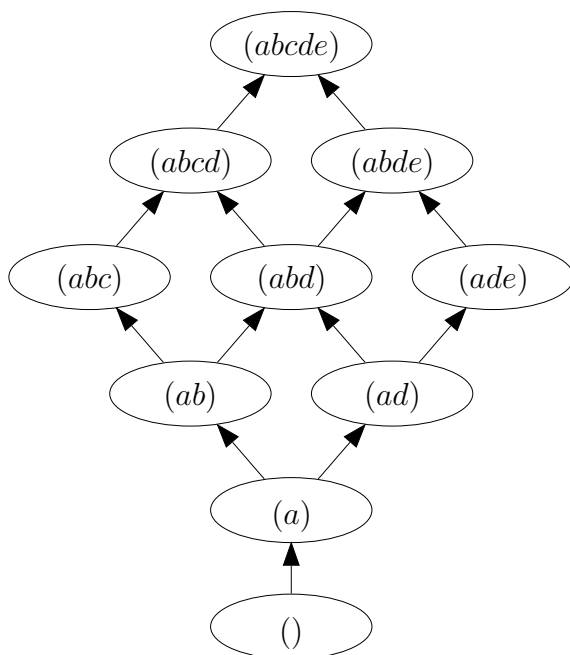


Figure 4.19 – Treillis $\mathcal{T}(G, a)$ correspondant au graphe G présenté sur la figure 4.18.

Ajout d'un nœud x (par une arête (x', x)).

Soit G' le graphe obtenu en ajoutant un nœud x à un graphe G par une arête (x', x) , avec $x' \in G$. L'algorithme 4 permet de construire $\mathcal{T}(G', a)$ à partir de $\mathcal{T}(G, a)$. L'algorithme commence par créer tous les états ayant pour successeur un nouvel état, puis crée les successeurs de ces nouveaux états.

La figure 4.20 présente le treillis $\mathcal{T}(G', a)$ obtenu sur un exemple, G étant le graphe de la figure 4.18, avec pour nouveau nœud $x = f$, relié à $x' = d$. On peut remarquer que tous les états de $\mathcal{T}(G, a)$ contenant d sont reliés à un nouvel état

Algorithme 4 : Algorithme de mise à jour du treillis lors de l'ajout d'un nœud x par une arête (x', x) .

Entrées : G est le graphe initial, a la graine, $\mathcal{T}(G, a)$ est le treillis correspondant à G , x est le nouveau nœud, $x' \in G$ est le nœud auquel x est connecté

$\mathcal{T}(G', a) \leftarrow \mathcal{T}(G, a)$

pour chaque $E \in \mathcal{T}(G, a)$ **tel que** $x' \in E$ **faire**

 | créer l'état $E \cup \{x\}$ dans $\mathcal{T}(G', a)$

 | ajouter $E \cup \{x\}$ comme successeur de E dans $\mathcal{T}(G', a)$

finpourchaque

pour chaque $E \in \mathcal{T}(G, a)$ **tel que** $x' \in E$ **faire**

 | **pour chaque** successeur E' de E dans $\mathcal{T}(G, a)$ **faire**

 | ajouter $E' \cup \{x\}$ comme successeur de $E \cup \{x\}$ dans $\mathcal{T}(G', a)$

 | **finpourchaque**

finpourchaque

retourner $\mathcal{T}(G', a)$

contenant f (il s'agit de la première boucle de l'algorithme). Par exemple, à l'état (ad) de $\mathcal{T}(G, a)$ est associé l'état (adf) de $\mathcal{T}(G', a)$. Les successeurs de (ad) dans $\mathcal{T}(G, a)$, c'est-à-dire (abd) et (ade) , correspondent aux successeurs $(abdf)$ et $(adef)$ de (adf) dans $\mathcal{T}(G', a)$ (il s'agit de la deuxième boucle de l'algorithme).

Suppression d'un nœud x .

Soit G' le graphe obtenu en supprimant le nœud $x \neq a$ au graphe G , en supposant que G' reste connexe²³. Le treillis $\mathcal{T}(G', a)$ est égal au treillis $\mathcal{T}(G, a)$ dans lequel tous les états contenant x ont été supprimés.

Ajout d'une arête (x, y) .

Soit G' le graphe obtenu en ajoutant l'arête (x, y) entre $x \in G$ et $y \in G$. L'algorithme 5 permet de construire $\mathcal{T}(G', a)$ à partir de $\mathcal{T}(G, a)$.

La figure 4.21 présente le treillis $\mathcal{T}(G', a)$ obtenu sur un exemple, G étant le graphe de la figure 4.18, pour l'ajout de l'arête (x, x') , avec $x = c$ et $y = e$. Les états de $\mathcal{T}(G, a)$ contenant $x = c$ mais pas $y = e$ sont (abc) et $(abcd)$. On ajoute le nouvel état $(abce)$ comme successeur de (abc) , et on remarque que $(abcde)$ était déjà un successeur de $(abcd)$. Les états de $\mathcal{T}(G, a)$ ne contenant pas $x = c$ mais contenant $y = e$ sont (ade) et $(abde)$. On ajoute le nouvel état $(acde)$ comme successeur de (ade) , et on remarque que $(abcde)$ était déjà un successeur de $(abcd)$. Pour obtenir les successeurs du nouvel état $(abce)$, on ajoute e aux successeurs de (abc) (le seul successeur était $(abcd)$) : on obtient $(abcde)$ comme seul successeur de $(abce)$. Finalement, pour obtenir les successeurs du nouvel état $(acde)$, on ajoute c aux successeurs de (ade) (le seul successeur était $(abde)$) : on obtient $(abcde)$ comme seul successeur de $(acde)$.

23. Si G' n'est pas connexe, il faut appliquer l'algorithme de suppression à tous les nœuds isolés.

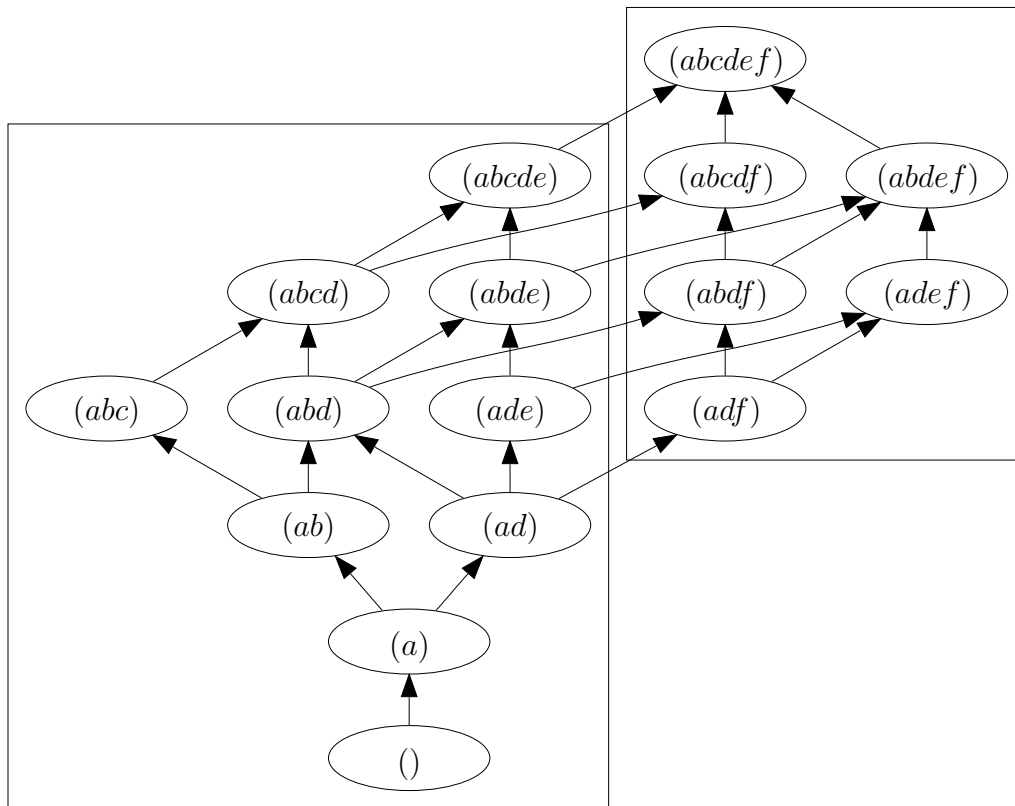


Figure 4.20 – Construction de $\mathcal{T}(G', a)$ à partir de $\mathcal{T}(G, a)$ lors de la mise à jour de G (voir figure 4.18) par l'ajout d'un nœud $x = f$ et d'une arête (x', x) , avec $x' = d$.

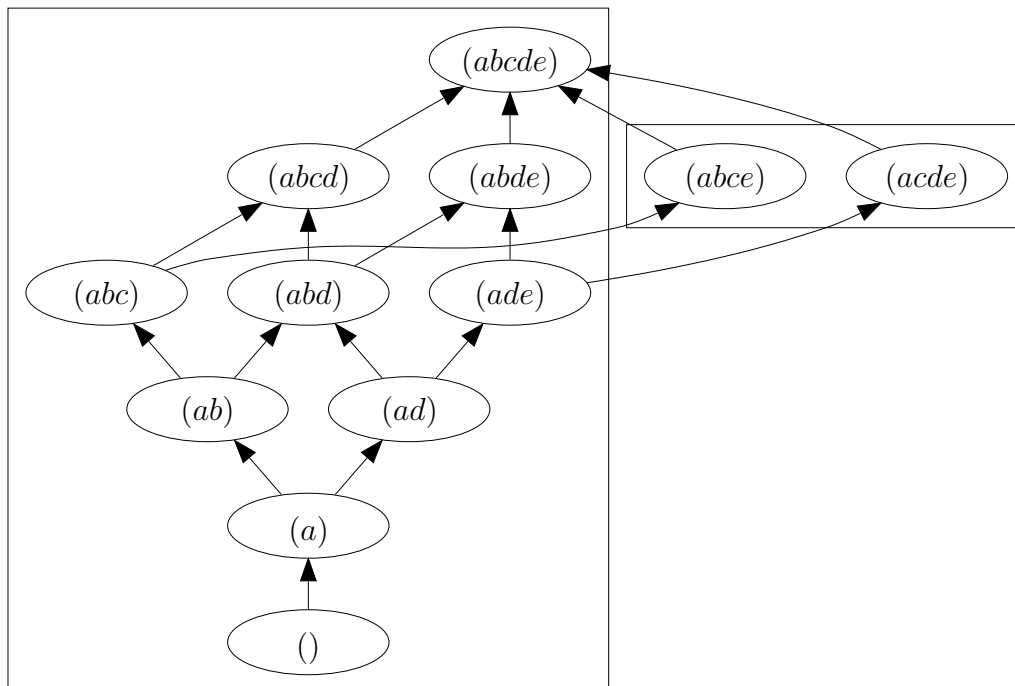


Figure 4.21 – Construction de $\mathcal{T}(G', a)$ à partir de $\mathcal{T}(G, a)$ lorsque G' est obtenu en ajoutant l'arête (c, e) à G (voir figure 4.18).

Algorithme 5 : Algorithme de mise à jour du treillis lors de l'ajout d'une arête (x, y) .

Entrées : G est le graphe initial, a la graine, $\mathcal{T}(G, a)$ est le treillis correspondant à G , $x \in G$ et $y \in G$ sont deux nœuds non reliés dans G

$\mathcal{T}(G', a) \leftarrow \mathcal{T}(G, a)$

pour chaque $E \in \mathcal{T}(G, a)$ **tel que** $x \in E$ **et** $y \notin E$ **faire**

si $E \cup \{y\}$ *n'existe pas dans* $\mathcal{T}(G', a)$ **alors**

 | créer $E \cup \{y\}$ dans $\mathcal{T}(G', a)$

fin

 ajouter $E \cup \{y\}$ comme successeur de E dans $\mathcal{T}(G', a)$

finpourchaque

pour chaque $E \in \mathcal{T}(G, a)$ **tel que** $x \notin E$ **et** $y \in E$ **faire**

si $E \cup \{x\}$ *n'existe pas dans* $\mathcal{T}(G', a)$ **alors**

 | créer $E \cup \{x\}$ dans $\mathcal{T}(G', a)$

fin

 ajouter $E \cup \{x\}$ comme successeur de E dans $\mathcal{T}(G', a)$

finpourchaque

pour chaque $E \in \mathcal{T}(G, a)$ **tel que** $x \in E$ **et** $y \notin E$ **faire**

pour chaque *successeur* E' **de** E **dans** $\mathcal{T}(G, a)$ **faire**

si $E' \cup \{y\}$ *n'est pas successeur de* $E \cup \{y\}$ **dans** $\mathcal{T}(G', a)$ **alors**

 | ajouter $E' \cup \{y\}$ comme successeur de $E \cup \{y\}$ dans $\mathcal{T}(G', a)$

fin

finpourchaque

finpourchaque

pour chaque $E \in \mathcal{T}(G, a)$ **tel que** $x \notin E$ **et** $y \in E$ **faire**

pour chaque *successeur* E' **de** E **dans** $\mathcal{T}(G, a)$ **faire**

si $E' \cup \{x\}$ *n'est pas successeur de* $E \cup \{x\}$ **dans** $\mathcal{T}(G', a)$ **alors**

 | ajouter $E' \cup \{x\}$ comme successeur de $E \cup \{x\}$ dans $\mathcal{T}(G', a)$

fin

finpourchaque

finpourchaque

retourner $\mathcal{T}(G', a)$

Suppression d'une arête (x, y) (ne déconnectant pas le graphe).

Soit G' le graphe obtenu en supprimant l'arête (x, y) de G , cette suppression ne déconnectant pas le graphe. L'algorithme 6 permet de construire $\mathcal{T}(G', a)$ à partir de $\mathcal{T}(G, a)$.

Algorithme 6 : Algorithme de mise à jour du treillis lors de la suppression d'une arête (x, y) ne déconnectant pas le graphe G' .

Entrées : G est le graphe initial, a la graine, $\mathcal{T}(G, a)$ est le treillis correspondant à G , $x \in G$ et $y \in G$ sont deux nœuds reliés dans G tels que la suppression de (x, y) ne déconnecte pas le graphe

$\mathcal{T}(G', a) \leftarrow \mathcal{T}(G, a)$

pour chaque $E \in \mathcal{T}(G, a)$ **tel que** $x \in E$ **et** $y \in E$ **faire**

si le sous-graphe de G' induit par E n'est pas connexe **alors**

 supprimer l'état E dans $\mathcal{T}(G', a)$

fin

finpour chaque

retourner $\mathcal{T}(G', a)$

La figure 4.22 présente le treillis $\mathcal{T}(G', a)$ obtenu sur un exemple, G étant le graphe de la figure 4.18, avec $x = a$ et $y = b$. On remarque que les états (ab) et (abc) de $\mathcal{T}(G, a)$ ont été supprimés : avec la suppression de l'arête (a, b) , ces deux états sont les seuls pour lesquels le sous-graphe induit dans G' n'est pas connexe.

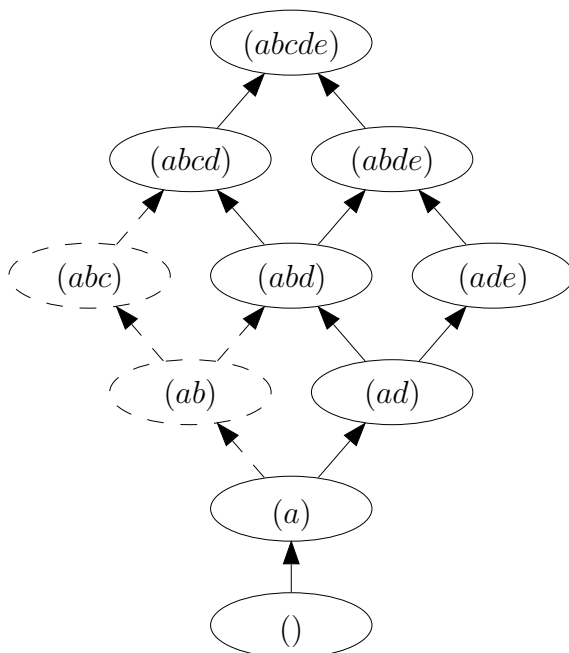


Figure 4.22 – Construction de $\mathcal{T}(G', a)$ à partir de $\mathcal{T}(G, a)$, lorsque G' est obtenu en supprimant l'arête (a, b) de G (voir figure 4.18).

4.3.2.3 Estimation du temps de transition d'un état à un autre

Nous allons à présent proposer un algorithme permettant d'estimer le temps de transition de l'état 1 à l'état 2, à partir de la notification initiale de changement d'état. Si l'on note a la graine, nous cherchons à estimer le temps maximum nécessaire pour que le réseau passe de l'état (a) à l'état \top .

Selon les hypothèses faites, il existe différentes manières de calculer le temps maximum pour passer de l'état (a) à l'état \top .

L'hypothèse la plus simple est de supposer qu'il existe un temps maximum $d(x, y)$ pour qu'un nœud x qui vient de passer de l'état 1 à l'état 2, informe le nœud y de passer de l'état 1 à l'état 2. Nous supposons que $d(x, y)$ est connu, et nous pouvons ainsi pondérer chaque arête (x, y) de G par $d(x, y)$. En notant $P = (x_0 = a, x_1, \dots, x_m)$ le plus long des plus courts chemins partant de a dans G , on peut montrer que le temps maximum pour passer de l'état (a) à l'état \top est $\sum_{i=0}^{m-1} d(x_i, x_{i+1})$. En effet, avec cette hypothèse, après un temps $\sum_{i=0}^{m-1} d(x_i, x_{i+1})$, tous les nœuds qui sont à une distance inférieure à $\sum_{i=0}^{m-1} d(x_i, x_{i+1})$ sont informés. Comme P est le plus long des plus courts chemins, il n'existe aucun nœud x' plus éloigné de a que x_m . On a donc la propriété attendue. Avec cette hypothèse, il n'est pas nécessaire d'utiliser le treillis.

La figure 4.23 présente la pondération $d(x, y)$ de chaque arête (x, y) pour un graphe G et une graine a . Le plus long chemin partant de a est le chemin (a, b, c) , de longueur 2. En effet, si le nœud a passe dans l'état 2 au temps 0, les nœuds b et d passent dans l'état 2 au plus tard au temps 1, et le nœud c passe dans l'état 2 au plus tard au temps 2.

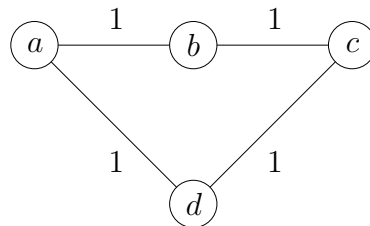


Figure 4.23 – Un exemple de graphe où est représenté sur chaque arête (x, y) le temps maximum pour que le nœud x informe le nœud y de changer d'état (ce qui correspond à une hypothèse simplificatrice).

L'hypothèse que le temps maximum $d(x, y)$ est simple à obtenir est en fait discutable. En effet, si l'on reprend l'exemple de la figure 4.23, on remarque qu'une fois que b et d sont dans l'état 2, ils sont en compétition pour informer le nœud c . Cette compétition peut se traduire par un retard, qui augmente le temps pour que c soit informé.

Présentons à présent une hypothèse plus générale (et permettant une plus grande précision), pour laquelle le temps maximum pour qu'un nœud y passe dans l'état 2 dépend de l'état de tous les autres nœuds (incluant ses voisins). Notons $d(E, y)$ le temps maximum pour qu'un ensemble E de nœuds dans l'état 2 informe un nœud y de passer de l'état 1 à l'état 2, avec y un voisin d'au moins un nœud de E . Pour chaque état E de ce treillis, tel que $a \in E$, on peut pondérer les arêtes de E aux

successeurs de E avec le temps $d(E, y)$ (où $E \cup \{y\}$ décrit l'ensemble des successeurs de E). En profitant du fait que $\mathcal{T}(G, a)$ est un graphe orienté sans cycle, on peut calculer en temps linéaire par rapport au nombre de nœuds dans le treillis (qui peut être exponentiel en fonction de la taille de G) le temps maximum pour passer de l'état (a) à l'état \top .

Un algorithme similaire peut être utilisé pour calculer le temps minimum de transition d'un état à un autre, à partir de la connaissance du temps minimum de transition d'un état E à un successeur E' dans le treillis. De même, un algorithme similaire peut être utilisé pour calculer le temps moyen de transition d'un état à un autre, à partir de la connaissance du temps moyen de transition d'un état E à un successeur E' dans le treillis, et de la probabilité de passer de E à E' .

4.3.2.4 Calcul de l'intervalle des états possibles

Le graphe d'évolution permet de calculer l'ensemble des états possibles du système à partir d'une connaissance de l'état de certains nœuds seulement. Étant donné que le graphe d'évolution forme un treillis, l'ensemble des états possibles du système peut être représenté comme un intervalle, au moyen d'un état Min et d'un état Max .

Soit G un graphe, a une graine, X un ensemble de nœuds qui sont connus pour être dans l'état 2, et \bar{X} un ensemble de nœuds qui sont connus pour être dans l'état 1 (avec $X \cap \bar{X} = \emptyset$). Il peut exister des nœuds de G qui ne sont ni dans X , ni dans \bar{X} : l'état de ces nœuds est alors inconnu. X et \bar{X} peuvent être vides. En revanche, si $X \neq \emptyset$, alors $a \in X$. De plus, si $a \in \bar{X}$, alors $X = \emptyset$.

L'état Max correspond à l'état où un maximum de nœuds sont dans l'état 2, à l'exception des nœuds de \bar{X} . L'état Max correspond à l'état \perp si $a \in \bar{X}$. Sinon, l'état Max est obtenu en calculant la composante connexe de $G \setminus \bar{X}$ contenant la graine a .

L'état Min correspond à l'état où un minimum de nœuds sont dans l'état 2. L'état Min correspond à l'état \perp si $a \notin X$. Sinon, l'état Min est obtenu en recherchant le plus petit graphe connexe contenant $X \cup \{a\}$. Les propriétés du treillis (et notamment le fait que chaque état du treillis est un graphe connexe) permettent de calculer l'état Min rapidement.

L'algorithme 7 permet de calculer cet état Min . Sa complexité est en $\mathcal{O}(n^2)$. En effet, la boucle principale de l'algorithme est exécutée au plus n fois (la hauteur du treillis étant égale à $n + 1$, avec n le nombre de nœuds du graphe), et la recherche d'un prédécesseur qui contient $X \cup \{a\}$ peut se faire en $\mathcal{O}(n)$, en considérant un à un les états $E \setminus \{x_i\}$, avec $x_i \in E \setminus X \setminus \{a\}$. Ainsi, la complexité de l'algorithme est $\mathcal{O}(n^2)$.

Il est intéressant de remarquer que l'état Min construit est le plus petit graphe connexe contenant a et tous les nœuds de X , ce qui correspond à un arbre de Steiner²⁴ couvrant $\{a\} \cup X$. En d'autres termes, l'algorithme 7 permet de construire

24. Un arbre de Steiner est un arbre couvrant partiel de poids minimum [Hak71]. Si on connaît l'ensemble des nœuds couverts par l'arbre de Steiner de $\{a\} \cup X$, c'est-à-dire les nœuds de l'état Min , il suffit de calculer un arbre couvrant total de poids minimum [Kru56, Pri57] pour obtenir l'arbre de Steiner de $\{a\} \cup X$. La construction d'un arbre de Steiner est un problème NP-complet [Kar72].

Algorithme 7 : Calcul de l'état *Min* à partir d'une connaissance partielle des états dans l'état 2.

Entrées : G est le graphe initial, a une graine, $\mathcal{T}(G, a)$ est le treillis correspondant à G , X est un ensemble de nœuds de G

$fin \leftarrow$ faux
 $E \leftarrow \top$

tantque non fin faire

si il existe un prédécesseur E' à E dans $\mathcal{T}(G, a)$ tel que $X \cup \{a\} \subset E'$

alors

$E \leftarrow E'$

sinon

$fin \leftarrow$ vrai

finsi

fintantque

retourner E

un arbre de Steiner en temps polynomial (en fonction du nombre de nœuds de $\mathcal{T}(G, a)$, qui peut être exponentiel par rapport au nombre de nœuds de G). Pour toutes les classes de graphes G tels que $\mathcal{T}(G, a)$ a un nombre d'états polynomial en $|G|$, notre algorithme permet donc de résoudre le problème de Steiner en temps polynomial.

4.3.3 Bilan de l'étude sur l'évolution du réseau

Nous venons de voir qu'un réseau de capteurs sans fil, suite à la détection et à la propagation d'un évènement significatif, évolue en suivant une structure de treillis. L'étude des treillis étant l'une des thématiques de l'axe 1 du LIMOS, il semble très prometteur de poursuivre les activités de recherche sur les particularités de ce graphe d'évolution d'un réseau de capteurs sans fil. Cette activité de recherche pourrait conduire à un rapprochement de l'axe 1 (modèles et algorithmes de l'aide à la décision) et de la partie réseaux de l'axe 2 (systèmes d'information et de communication) du LIMOS.

Le fait que le graphe d'évolution du réseau soit un treillis a des conséquences utiles pour les concepteurs de protocoles. Grâce aux propriétés du treillis, il est possible d'estimer le temps de transition d'un état à un autre, lorsque ce temps prend en compte le comportement de la sous-couche MAC (voir l'algorithme 4.3.2.3). Le temps de transition correspond au temps maximum pour aller du graphe où seule la graine est dans l'état 2, au graphe où tous les nœuds sont dans l'état 2. La recherche du plus long chemin dans le treillis peut être faite rapidement (une fois que le treillis est construit). Rappelons que ce temps est utilisé pour paramétrer l'automate des nœuds, afin de garantir que les nœuds ne changent pas trop souvent d'état, ce qui complexifie la gestion du réseau (voir la partie 4.2.2).

Une autre utilité de la représentation de l'évolution du réseau par un treillis concerne les applications fonctionnant sur le réseau de capteurs sans fil. À partir d'une vision partielle du réseau, qui se traduit par la connaissance de l'état de certains des nœuds, il est possible pour une application de calculer l'ensemble des états

dans lequel le réseau peut se trouver (voir l'algorithme 4.3.2.4). Cette connaissance peut donner à l'application des informations supplémentaires sur les nœuds dont l'état est inconnu. Le fait que cet ensemble puisse être représenté comme un intervalle est une conséquence du fait que le graphe d'évolution est un treillis. Le calcul rapide de cet intervalle utilise aussi des propriétés du treillis.

Deux types de perspectives s'ouvrent donc à nous. Premièrement, nous pourrions approfondir notre connaissance du graphe d'évolution en cherchant de nouvelles propriétés. Ce travail pourrait être entamé par un étudiant en master recherche, par exemple. Deuxièmement, nous pourrions chercher à trouver d'autres utilisations des propriétés du treillis pour les concepteurs de protocoles réseaux, ou pour les applications fonctionnant sur le réseau de capteurs sans fil.

4.4 Conclusions

Dans un réseau de capteurs sans fil utilisant une architecture multi-piles, les protocoles réseaux (ou leur paramétrage) peuvent évoluer en fonction des événements détectés ou des commandes d'un opérateur. Nous montrons ce comportement au travers de l'exemple du projet Clervolc, présenté dans la partie 4.1. Dans cette application de surveillance environnementale d'un volcan, le réseau est en état de veille l'essentiel du temps, ce qui permet d'économiser de l'énergie. Dès qu'un séisme important est détecté, les nœuds du réseau basculent (de manière désynchronisée) dans un état de surveillance intensive, où le délai de transmission des paquets devient une métrique principale.

Dans les architectures multi-piles globales, étudiées dans la partie 4.2, le passage d'une pile protocolaire à l'autre est lié à un événement externe au réseau. De plus, ce changement de pile protocolaire est global : tous les nœuds doivent utiliser (à terme) la nouvelle pile protocolaire. Nous proposons des algorithmes permettant d'éviter les boucles pendant la phase transitoire où certains nœuds ont basculé vers la nouvelle pile protocolaire, tandis que d'autres nœuds ne sont pas encore avertis du changement de pile protocolaire.

Dans la partie 4.3, nous nous concentrons sur l'évolution de la pile protocolaire utilisée par les nœuds au cours du temps. Nous montrons que l'ensemble des états possibles du réseau forme une structure particulière, et nous en étudions ses propriétés, en faisant le lien entre la topologie du réseau et les évolutions possibles. Nous profitons de ces résultats pour proposer un algorithme permettant d'estimer la durée nécessaire pour que tous les nœuds du réseau aient basculé vers la nouvelle pile protocolaire (cet algorithme pouvant être utilisé par les concepteurs de protocoles pour paramétrer la durée de transition minimale entre deux états). Nous proposons aussi un algorithme permettant de déterminer l'intervalle des états possibles du réseau, à partir de la connaissance de la pile protocolaire utilisée par un sous-ensemble de nœuds (cet algorithme pouvant être utilisé par les applications du réseau de capteurs sans fil).

Ces travaux ouvrent de nombreuses pistes de recherche : la recherche d'autres exemples d'architectures globales, l'étude d'autres propriétés des architectures globales (ne concernant pas les boucles, mais le délai par exemple), l'étude des pro-

propriétés du graphe d'évolution, ou l'étude de la manière dont ces propriétés peuvent être utilisées pour aider les concepteurs de protocoles réseaux ou les applications du réseau.

CHAPITRE

5

RÉSUMÉ ET PERSPECTIVES

Dans ce chapitre, nous commençons par résumer les principales contributions présentées dans ce document. Ensuite, nous donnons un résumé de nos contributions qui ont été omises, soit par manque de place, soit pour rester concentré sur la contribution principale. Ensuite, nous faisons le lien entre les thématiques de recherche et les étudiants de thèse et de master recherche que j'ai encadrés (ou co-encadrés). Finalement, nous présentons plusieurs perspectives à nos travaux.

5.1 Résumé des contributions présentées

Nous avons montré que les réseaux de capteurs sans fil pouvaient être utilisés pour de nombreuses applications de surveillance de sites étendus, notamment industriels ou environnementaux. Nous nous sommes placés dans le cas où plusieurs piles protocolaires sont utilisées dans le même réseau de capteurs sans fil. Ce cas de figure se présente dans deux scénarios complémentaires. Dans le premier scénario, les qualités de service requises par l'application ne peuvent pas être fournies par une seule pile protocolaire. Ce scénario se produit lorsque l'application est susceptible de générer des trafics très différents, par exemple un trafic de suivi de basse priorité et un trafic d'alarmes de haute priorité. Dans le deuxième scénario, de nombreuses applications coexistent dans un même déploiement réseau. Ce scénario permet de réduire les coûts liés au déploiement, puisque les nœuds capteurs sont mutualisés par les applications. En revanche, le trafic applicatif soumis au réseau augmente avec le nombre d'applications. Le nombre de piles protocolaires que le réseau doit pouvoir gérer augmente lui aussi avec le nombre d'applications, même si certains protocoles peuvent être réutilisés et mutualisés.

Dans le chapitre 3, nous avons présenté les protocoles proposés dans le cadre

du projet Ocari. Ces protocoles peuvent être utilisés pour la surveillance d'un site industriel, et permettent de fournir deux qualités de service très différentes, tout en profitant de nombreuses optimisations utilisant des techniques de *cross-layering*. Nous avons montré que ces protocoles pouvaient s'intégrer dans une architecture multi-piles, et nous avons mis en avant deux types d'architectures multi-piles : les architectures simples et les architectures locales. Nous avons introduit des propriétés de compatibilité et de retardabilité sur les combinaisons de protocoles de routage (et non pas sur les protocoles de routage eux-mêmes), qui permettent d'éviter les boucles de routage qui seraient susceptibles d'apparaître dans les architectures locales.

Dans le chapitre 4, nous avons présenté les protocoles proposés dans le cadre du projet Clervolc. Ces protocoles peuvent être utilisés pour la surveillance d'un site environnemental, et permettent de fournir des qualités de service qui dépendent des événements détectés. Nous avons montré que ces protocoles s'intégraient dans un nouveau type d'architectures multi-piles : les architectures globales. Nous avons proposé des protocoles permettant de gérer la phase de transition pendant laquelle certains nœuds utilisent une nouvelle pile protocolaire (suite à la détection d'un événement jugé significatif par l'application), tandis que d'autres utilisent encore l'ancienne pile protocolaire. Finalement, nous avons montré que pendant ces phases de transition, le réseau évoluant d'une manière qui peut être modélisée par un treillis, et nous avons mis en avant quelques propriétés de ce treillis. Nous avons donné, notamment, un algorithme de calcul de la durée maximale de la phase de transition, ainsi qu'un algorithme de calcul d'un intervalle d'états possibles du réseau à partir de la connaissance de l'état de certains nœuds seulement.

Pour résumer, nous avons présenté dans ce document nos travaux de recherche dont la thématique générale porte sur les architectures multi-piles, en insistant sur l'étude des propriétés des combinaisons de protocoles de routage.

5.2 Résumé des contributions non présentées

Dans cette partie, nous présentons rapidement quelques contributions qui n'ont pas été présentées dans le cœur de ce document. Nous avons choisi de présenter trois aspects¹ : la démarche de simulation réaliste, le routage tolérant au délai, et la réduction de la congestion.

5.2.1 Démarche de simulation réaliste

La démarche de simulation, même si elle n'est pas présentée en détail dans ce document, est centrale dans l'évaluation des solutions proposées par l'ensemble de l'équipe « réseaux et protocoles ». Même s'il a été montré que différents simulateurs donnaient généralement des résultats différents [CSS02], nous pensons qu'il est possible d'affiner les simulateurs pour que les résultats qu'ils produisent soient réalistes.

1. Trois autres aspects de nos travaux, plus mineurs, ne sont ni présentés dans le cœur de ce document, ni dans cette conclusion : il s'agit de l'optimisation de requêtes multiples dans un réseau de capteurs sans fil, la compression de séries temporelles représentant les données issues d'un réseau de capteurs sans fil, et l'évaluation et l'optimisation de la sous-couche MAC de IEEE 802.15.4.

Notre démarche de simulation réaliste peut être décrite en trois étapes : l'implémentation d'un simulateur réaliste, la mise en place d'un environnement complet de simulation, et le retour d'informations des résultats de prototypes pour la simulation.

L'implémentation d'un simulateur réaliste est le point de départ de notre démarche de simulation. Nous nous sommes basés sur le simulateur réseau NS2 (version 2.31), très utilisé dans la communauté scientifique internationale. Nous avons réécrit l'essentiel des couches basses de NS2, afin de prendre en compte des aspects qui étaient négligés dans la version originale. Nous pouvons citer l'implémentation du modèle de propagation ITU en intérieur [ITU 1238], la prise en considération des variations des conditions de propagation d'un lien donné (qui permet d'avoir, pour une distance donnée, des liens de bonne qualité et des liens de mauvaise qualité, ce qui est le cas en pratique [BARD13]), la prise en compte de la dérive temporelle des nœuds, l'amélioration de la modélisation de la couche physique (avec la prise en compte de la probabilité d'erreurs par bit due à la modulation, l'utilisation de huit tests différés de médium pour réaliser un CCA (comme le standard IEEE 802.15.4 l'impose) plutôt qu'un test unique et immédiat, ou encore la prise en compte de l'effet de capture temporel), et l'introduction d'un temps de traitement pour certaines tâches.

La mise en place d'un environnement complet de simulation permet d'utiliser efficacement notre simulateur. Cet environnement comprend deux types d'outils logiciels : des scripts de préparation des simulations, et des scripts de collecte des résultats. Les scripts de préparation des simulations gèrent tous les paramètres des protocoles testés et génèrent les topologies (qu'il s'agisse de topologies en lignes, en grilles, en arbres, aléatoires uniformes, ou aléatoires non uniformes). Ces scripts s'occupent aussi du lancement du simulateur pour chacune des répétitions prévues, et sauvegardent les résultats (à la fois les fichiers de trace et la sortie du simulateur). Ces scripts permettent de garantir que les résultats produits correspondent bien à la dernière version des paramètres. Les scripts de collecte des résultats, quant à eux, extraient les valeurs des métriques utiles à partir de tous les fichiers de résultats produits, et produisent des fichiers contenant des valeurs numériques ou des courbes. Ces scripts réalisent eux-mêmes les calculs de moyennes et d'intervalles de confiance. L'utilisation de ces scripts permet de minimiser les erreurs d'extraction, et accélère significativement la mise en page des résultats produits.

Le retour d'informations des résultats des prototypes pour l'intégration dans notre simulateur permet d'affiner régulièrement le réalisme de notre simulateur. D'ailleurs, ces retours d'informations sont nécessaires dans une démarche de simulation visant le réalisme [PTBD10]. Ils ont conduit de manière directe aux publications suivantes [GTH08*, CLG⁺09*, ABD⁺13*], et de manière indirecte à une bonne partie des publications de l'équipe.

Cette démarche de simulation, créée autour d'outils logiciels, s'appuie aussi sur des contributions scientifiques permettant de réaliser certains points spécifiques. Par exemple, nous avons proposé des algorithmes pour accélérer l'association des nœuds au réseau (qui est souvent une phase obligatoire mais annexe lorsque l'on teste des protocoles MAC ou réseau) [ERGM09*, ERGM12c*], ou pour simplifier les problèmes d'adressage sur certaines topologies aléatoires [ERGBM10*].

5.2.2 Routage tolérant au délai

Le routage tolérant au délai consiste à proposer des mécanismes de routage applicables sur des réseaux où les liens sont intermittents. Dans de tels réseaux, la connectivité n'est pas garantie à un instant donné.

L'une des contributions de la thèse de Nassima HADID [Had11] a été de proposer un protocole de routage tolérant au délai pour une application de robotique mobile. Pour gérer les différents capteurs et actionneurs des robots autonomes et mobiles, nous avons remarqué qu'il est possible d'utiliser un réseau de capteurs sans fil fonctionnant selon la norme IEEE 802.15.4. Dans le cas où les robots doivent interagir, il est nécessaire d'introduire de nouveaux protocoles. Pour ne pas perdre la compatibilité avec la norme IEEE 802.15.4, nous avons proposé une architecture dans laquelle les nœuds du réseau qui sont riches en énergie profitent de l'inactivité des autres nœuds (qui sont pauvres en énergie) pour communiquer entre eux. Ces nœuds riches en énergie se placent alors sur un canal commun à tous les mobiles, appelé canal d'aparté, et interagissent, sans que la compatibilité avec la norme IEEE 802.15.4 ne soit remise en cause [HGM11a*, HGM11b*]. Plusieurs aspects de cette architecture correspondent à un réseau tolérant au délai. En effet, les robots sont mobiles et ne sont en contact radio que rarement. De plus, même lorsque les robots sont à portée radio, ils ne peuvent communiquer entre eux que lorsque le réseau de capteurs de chacun des robots est dans la phase d'activité sur le canal d'aparté, c'est-à-dire que les nœuds pauvres en énergie sont dans leur phase d'inactivité. Pour gérer les communications dans un tel réseau, nous avons mis en place un protocole de routage basé sur une structure appelée *bundle*, servant à temporiser les paquets et permettant un routage opportuniste.

L'une des études à mener pendant la thèse d'Affoua Thérèse ABY concerne aussi la proposition d'un protocole de routage tolérant au délai pour une application de surveillance environnementale. Le très faible taux d'activité des nœuds rend les liens intermittents, et rend par conséquent la topologie dynamique, même en l'absence de mobilité des nœuds. La conception de protocoles de routage supportant ces intermittences font partie des problématiques étudiées actuellement par Thérèse.

Il est probable que des protocoles tolérants au délai soient étudiés par de futurs étudiants en thèse ou en master. En effet, l'intermittence des liens apparaît de manière naturelle dans de nombreuses applications, dès lors que les nœuds ne sont pas synchronisés et que le taux d'activité de chaque nœud est faible.

5.2.3 Réduction de la congestion

Il est important de réduire la congestion afin d'améliorer les performances du réseau en termes de délai et de taux de pertes. Pendant la thèse de Nancy EL RACHKIDY, nous avons abordé ce point en remarquant que la congestion était souvent importante au niveau des sources dans un réseau de capteurs sans fil, notamment quand des situations critiques étaient détectées. En effet, il est probable que plusieurs nœuds capteurs dans une même zone détectent la même situation critique, dans un intervalle de temps court. Ces nœuds vont alors tous envoyer leurs données au puits, ce qui va provoquer de la congestion sur le chemin de la source au puits. Nous avons donc proposé un protocole basé sur des pivots afin d'écartier les

chemins des sources au puits [ERGBM09*, ERGM12b*, ERGB13*]. Une approche complémentaire pourrait être de chercher à réduire les interférences quand la position des nœuds est connue et que les interférences peuvent être estimées à partir de la distance entre les nœuds [HLP11].

Ensuite, et toujours avec Nancy EL RACHKIDY, nous avons cherché à réduire la congestion au niveau du puits. Cette congestion est inévitable lorsqu'un seul puits est dans le réseau. Nous avons donc étudié le cas où de nombreux puits sont déployés, ce qui nous a conduit à proposer un protocole supportant les communications *any-cast*, c'est-à-dire les communications pour lesquelles un ensemble de destinations équivalentes est fourni [ERGM10*, ERGM12a*]. Ce protocole combine la sélection d'un puits adapté et la construction d'une route écartée des autres routes afin de réduire la congestion.

5.3 Bilan des encadrements

La figure 5.1 présente les liens entre mes thématiques de recherche (représentées dans le cadre de deux contextes applicatifs : industriel et environnemental, correspondant aux projets Ocarl et Clervolc) et les étudiants de thèse et de master recherche que j'ai encadrés (ou co-encadrés). Généralement, les étudiants de thèse travaillent sur plusieurs thématiques, alors que les étudiants de master recherche ne se concentrent que sur une seule.

Dans les années à venir, il est probable que mes activités de recherche autour des applications environnementales (et par conséquent, les activités des étudiants que j'encadre) prennent de l'ampleur, étant donné que les projets en cours de l'équipe concernent surtout ce domaine applicatif. Comme plusieurs problématiques de routage semblent émerger des projets sur lesquels nous sommes impliqués, il est probable que nous travaillions prochainement sur ce point, en nous concentrant sur le passage à l'échelle, le routage géographique, le routage tolérant au délai, ou encore le routage réactif dans un réseau dynamique.

5.4 Perspectives

Les travaux décrits dans ce document font apparaître de nombreuses perspectives, dont certaines sont développées dans cette partie.

5.4.1 Surveillance environnementale

La surveillance environnementale constitue une thématique principale et fédératrice pour l'équipe « réseaux et protocoles », et constitue donc une perspective importante pour les années à venir.

Le projet Clervolc, qui est actuellement le plus important projet de cette thématique, est un projet relativement récent, et constitue en lui-même une perspective. Les thématiques de recherche actuellement abordées dans ce projet concernent plusieurs aspects : la partie matérielle des nœuds capteurs, l'affinage des protocoles

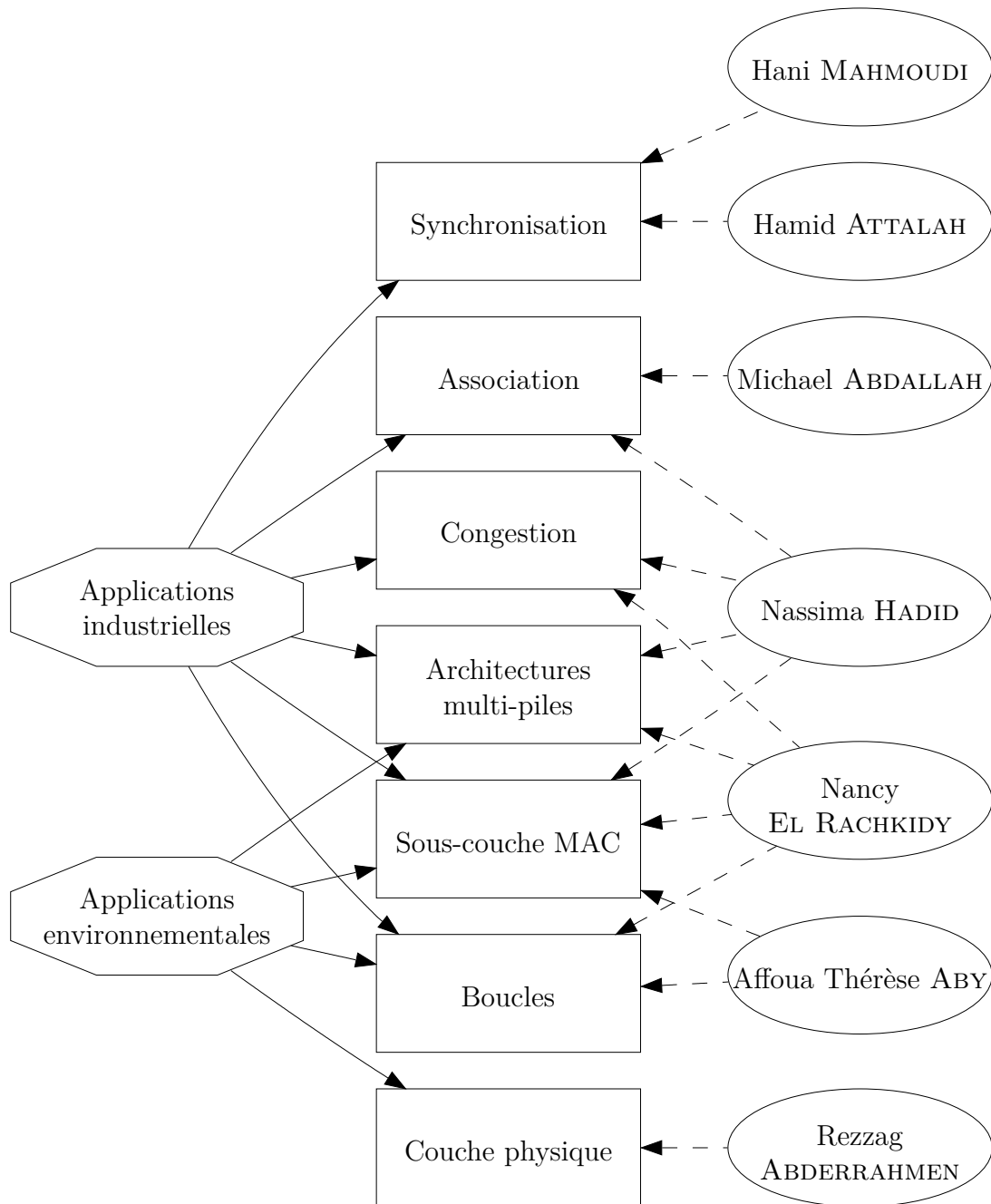


Figure 5.1 – Liens entre mes projets de recherche (représentés dans des polygones), mes problématiques de recherche (représentées dans des carrés) et mes encadrements (représentés dans des ellipses).

MAC et réseau (pour leur modélisation, leur simulation et leur validation), et l'expérimentation sur le terrain. Par rapport à la partie matérielle des nœuds capteurs, il faut prendre en compte la robustesse et la discrétion des nœuds capteurs, de leur source d'énergie, et de leur antenne (ce qui sera traité par l'intermédiaire de collaborations locales et canadiennes). De plus, il faut étudier la possibilité de disposer de plusieurs interfaces de transmissions radios, utilisant des technologies différentes et fonctionnant sur des bandes de fréquences adaptées. Par rapport à l'affinage des protocoles MAC et réseau, la gestion fine de la topologie, des performances réseaux et de l'énergie sont trois points qu'il faut étudier dans les protocoles développés. De plus, il est important de considérer des aspects tolérants au délai dans les protocoles de communication, afin de pouvoir prendre en compte des changements temporaires mais relativement longs de conditions de propagation (lorsqu'il pleut par exemple). Par rapport à l'expérimentation sur le terrain, plusieurs actions sont à entreprendre. Tout d'abord, il faut évaluer les conditions de déploiement des nœuds capteurs et les conditions de propagation dans un environnement réel. Ensuite, il faut implémenter un protocole applicatif complet permettant de réaliser des mesures expérimentales (avec l'envoi de commandes, le stockage des résultats, et le rapatriement des résultats).

En complément du projet Clervolc, nous cherchons à mettre en place un projet franco-canadien sur la réalisation de nœuds capteurs pouvant s'adapter à des déploiements rapides en extérieur, au niveau du sol. Nous envisageons d'intégrer les nœuds capteurs dans des dispositifs ayant une forme cubique de quelques centimètres de diamètre. La coque cubique des nœuds capteurs constituerait une antenne intelligente, pouvant modifier dynamiquement les lobes de son diagramme de rayonnement. Le nœud capteur en lui-même se situerait à l'intérieur de cette coque. Les communications auraient lieu au niveau du sol, ce qui dégrade énormément la qualité des communications. Nous nous concentrerions sur l'auto-organisation et l'auto-réparation du réseau de capteurs ainsi déployé.

Le projet AMI Géothermia est aussi une perspective importante. L'objectif de ce projet est d'équiper une station géothermique moderne d'un réseau de capteurs sans fil, pour surveiller la zone autour de la station afin de détecter des événements dangereux ou les impacts de la station sur l'écosystème. Les défis scientifiques consistent à transmettre quotidiennement des quantités d'informations volumineuses, tout en économisant l'énergie des nœuds capteurs. En plus des problématiques réseaux, il est important d'étudier différentes possibilités de compression du trafic produit par les nœuds capteurs.

Enfin, nous cherchons à initier des travaux sur la surveillance de touristes dans des sites naturels, afin d'améliorer leur sécurité. Les objectifs de l'application seraient d'assurer un suivi quotidien du réseau, de détecter des changements de topologie permanents, que ces changements soient majeurs (suite à la disparition de certains nœuds capteurs) ou mineurs (suite au déplacement sur quelques mètres d'un nœud capteur, causé par un petit glissement de terrain par exemple), voire de localiser des touristes en difficulté. Les défis scientifiques posés seraient de déployer des réseaux de capteurs sans fil robustes, pouvant pallier à la perte de certains nœuds, et de concevoir des protocoles passant à l'échelle, afin de pouvoir surveiller de grandes zones. La détection de changements stables dans les conditions de propagation pourrait être

utilisée pour mettre en avant des changements d’environnement.

5.4.2 Architectures multi-piles

Dans l’état de l’art de ce document, et en introduction des chapitres 3 et 4, nous avons présenté quelques architectures multi-piles concrètes. L’identification d’un plus grand nombre d’architectures multi-piles, ainsi que la classification de ces nouvelles architectures selon leur type (c’est-à-dire en architectures simples, locales et globales) est une perspective à court terme. Ce travail peut être mené en parallèle d’une réflexion sur le coût logiciel des architectures multi-piles, notamment en termes d’empreinte mémoire du code et de volume de données stockées. Le coût énergétique des architectures multi-piles peut aussi être évalué, afin de déterminer si elles permettent de faire des économies d’énergie ou non, et dans quels cas. Il est aussi possible d’étudier les liens entre les méthodes permettant d’éviter les boucles de routage transitoires proposées dans [FB07, CMP⁺13] et notre approche (voir la partie 2.1.2 pour une première discussion sur ce point).

Nous cherchons aussi à savoir s’il existe d’autres types d’architectures multi-piles (voire des sous-types). Par exemple, il pourrait s’agir d’architectures multi-piles pour lesquelles la décision de changement d’état est faite sur des critères globaux, mais pour lesquelles la décision n’est pas commune à tous les nœuds. Nous pourrions aussi envisager des architectures dans lesquelles plusieurs états incompatibles peuvent survenir (et persister) dans le réseau². Un autre exemple posant le même problème est la suppression du temps d’attente pour qu’un nœud puisse changer d’état : les nœuds pourraient évoluer très rapidement d’un état à un autre, indépendamment de la vitesse à laquelle les autres nœuds du réseau sont informés de ces changements. Dans une telle architecture, le réseau pourrait donc être en permanence dans un état non cohérent (par rapport à notre étude). Finalement, nous pourrions considérer des architectures multi-piles où la fusion des piles protocolaires n’est pas faite par la sous-couche MAC, mais par la couche réseau, par exemple.

Une direction complémentaire de recherche est l’étude plus avancée des combinaisons de protocoles, en considérant les protocoles de la couche réseau ou les protocoles de la sous-couche MAC. Concernant la couche réseau, nous pouvons chercher à adapter la notion de retardabilité (introduite pour les architectures locales dans la partie 3.3.2) aux architectures globales, en prenant garde au fait que les protocoles de routage ne sont plus alternés dans les architectures globales. Plutôt que d’utiliser le maximum des fonctions f_1 et f_2 , correspondant aux protocoles \mathcal{R}_1 et \mathcal{R}_2 , cette adaptation de la notion de retardabilité pourrait prendre en compte le fait que les voisins du nœud fonctionnent selon \mathcal{R}_1 ou \mathcal{R}_2 par exemple. Par ailleurs, nous avons montré que dans le cas des architectures globales, quand le protocole de routage \mathcal{R}_2 était interne, aucune boucle ne pouvait se produire (voir la partie 4.2.3.3). Nous pourrions chercher à déterminer s’il existe d’autres propriétés sur \mathcal{R}_2 (ou éventuellement sur \mathcal{R}_1) qui pourraient empêcher les boucles. Finalement, il serait intéressant de trouver des propriétés de combinaisons de protocoles de routage modélisant d’autres aspects

2. Rappelons que dans la partie 4.2.2, nous avons fait l’hypothèse que lorsque des nœuds évoluaient indépendamment vers des états différents (à cause de la détection d’évènements différents), les nœuds devaient à terme converger vers un état commun compatible.

que les boucles de routage, comme par exemple le nombre de sauts pour atteindre la destination. Concernant la sous-couche MAC, il serait intéressant d'approfondir la notion de p -rejoignabilité quand les protocoles ne sont pas indépendants. Des propriétés concernant le délai (plutôt que seulement la probabilité de communication) pourraient aussi être proposées pour les protocoles MAC de transition. Finalement, il serait intéressant de considérer la manière dont deux sous-couches MAC pourraient être exécutées de manière simultanée (plutôt que de manière séquentielle), et les avantages d'une telle approche.

Une autre perspective serait d'étudier les architectures multi-piles dans lesquelles la couche physique est multi-canaux, l'avantage étant une meilleure adaptabilité et un débit augmenté. Dans un premier temps, l'utilisation de plusieurs canaux pourrait être contrôlé par les sous-couches MAC de chacune des piles protocolaires. Il en découlerait alors des problématiques intéressantes de partage de canaux entre sous-couches MAC. Dans un deuxième temps, il serait intéressant de voir s'il est possible d'avoir plusieurs couches physiques différentes qui cohabitent dans une même architecture (par le biais de plusieurs modules radios distincts).

5.4.3 Étude du graphe d'évolution

Dans la partie 4.3, nous avons montré que l'étude des propriétés du graphe d'évolution du réseau pouvait aider l'administrateur du réseau à paramétrer la durée minimale entre deux changements d'état, ou à avoir une vision de l'état complet du réseau à partir d'une vision partielle. La compréhension plus fine des propriétés du graphe d'évolution peut apporter des réponses à d'autres questions.

De manière générale, deux types de résultats peuvent apparaître en continuant la recherche dans cette voie : de nouveaux résultats concernant les réseaux, et de nouveaux résultats concernant les treillis. Les nouveaux résultats concernant les réseaux auront probablement trait aux possibilités d'évolution du réseau, mais ils nécessiteront probablement que le treillis soit manipulable (c'est-à-dire, que son nombre d'états soit raisonnable ou qu'il puisse être représenté partiellement). Les nouveaux résultats concernant les treillis peuvent provenir du fait que l'approche réseau donne de nouveaux axes d'études, par exemple en identifiant les relations entre des nœuds particuliers de la topologie (ou des sous-graphes particuliers de la topologie), et certains états du treillis.

Les résultats peuvent concerner plusieurs aspects : les propriétés de certains états, les propriétés du treillis, et la représentation du treillis en fonction de différentes hypothèses (évolution unitaire ou multiple, évolution mono-graine ou multi-graines, évolution bi-états ou multi-états, ou encore topologie connexe ou non). Concernant les propriétés de certains états, on peut chercher à dénombrer le nombre de sup-irréductibles, ou à déterminer les états ayant des compléments ou des pseudo-compléments. Concernant les propriétés du treillis, on peut chercher à identifier la structure du treillis quand la topologie du réseau est particulière. Dans le cas où la topologie du réseau est quelconque, on peut chercher à connaître le nombre d'états du treillis ou le nombre d'états par niveau du treillis. Il serait intéressant de déterminer les classes de topologies produisant des treillis de taille polynomiale. Il serait aussi intéressant de déterminer si le treillis a toujours une forme de diamant (dans

le sens où le nombre d'états du treillis augmente jusqu'à un certain niveau, puis diminue), ce qui est actuellement notre intuition. Concernant la représentation du treillis, on peut chercher à construire (efficacement) des représentations partielles du treillis (plus facilement manipulables que le treillis complet), ou à identifier des sous-graphes du treillis qui permettraient de reconstruire facilement le treillis complet (idéalement, en trouvant des relations entre les sous-graphes de la topologie et les sous-graphes du treillis). Nous pouvons aussi approfondir notre connaissance des relations entre topologie et treillis, en cherchant à déterminer quels graphes produisent le même treillis, ou quels treillis correspondent à des graphes.

On peut finalement rappeler que l'étude des propriétés des treillis est une thématique importante de l'axe 1 (modèles et algorithmes de l'aide à la décision) du LIMOS, et notamment du groupe travaillant sur les algorithmes, les graphes et la complexité. L'étude du graphe d'évolution d'un réseau pourrait donc permettre d'accroître les activités de recherche inter-axes.

RÉFÉRENCES

- [AACG⁺09*] K. AL AGHA, G. CHALHOUB, A. GUITTON, E. LIVOLANT, S. MAHFOUDH, P. MINET, M. MISSON, J. RAHMÉ, T. VAL et A. VAN DEN BOSSCHE : Cross layering in an industrial wireless sensor network: Case study of Ocari. *JNW (Journal of Networks)*, 6(4), décembre 2009. Numéro spécial sur Wireless sensor networks: theory and practice.
- [ABD⁺13*] D. ABRIGNANI, C. BURATTI, D. DARDARI, N. EL RACHKIDY, A. GUITTON, F. MARTELLI, A. STAJKIC et R. VERDONE : The EuWIn testbed for 802.15.4/ZigBee networks: From the simulation to the real-world. Dans *ISWCS (International Symposium on Wireless Communication Systems)*, 2013.
- [ABJ08] M. ALI, A. BÖHM et M. JONSSON : Wireless sensor networks for surveillance applications – a comparative survey of MAC protocols. Dans *ICWMC (International Conference on Wireless and Mobile Communications)*, pages 399–403, 2008.
- [AGM13a*] A. T. ABY, A. GUITTON et M. MISSON : MAC mechanism for a scalable wireless sensor network using independent duty cycles. Dans *NICST (International France-China Workshop on New and Smart Information Communication Science and Technology to Support Sustainable Development)*, 2013.
- [AGM13b*] A. T. ABY, A. GUITTON et M. MISSON : Unsynchronized and scalable MAC protocol for wireless sensor networks. Travail en cours, 2013.
- [AKK04] J. N. AL-KARAKI et A. E. KAMAL : Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.
- [AMC⁺06] G.-S. AHN, E. MILLUZO, A. T. CAMPBELL, S. G. HONG et F. CUOMO : Funneling-MAC : A localized, sink-oriented MAC for boosting fidelity in sensor networks. Dans *SenSys (ACM Conference on Embedded Networked Sensor Systems)*, pages 293–306, 2006.
- [ATRD12] N. ABDEDDAIM, F. THEOLEYRE, F. ROUSSEAU et A. DUDA : Multi-channel cluster tree for 802.15.4 wireless sensor networks. Dans *PIMRC (IEEE*

- International Symposium on Personal, Indoor and Mobile Radio Communications*), 2012.
- [AY05] K. AKKAYA et M. YOUNIS : A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349, 2005.
- [BARD13] A. BILDEA, O. ALPHAND, F. ROUSSEAU et A. DUDA : Link quality metrics in large scale indoor wireless sensor networks. Dans *PIMRC (IEEE International Symposium on Personal, Indoor and Mobile Radio Communications)*, 2013.
- [BBB12] S. BOULFEKHAR, L. BOUALLOUCHE et M. BENMOHAMMED : S-TDMA: a new MAC protocol for wireless sensor networks. Dans *DS-RT (IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications)*, pages 29–35, 2012.
- [BHDK09] A. BACHIR, M. HEUSSE, A. DUDA et Leung K. K. : Preamble sampling mac protocols with persistent receivers in wireless sensor networks. *IEEE Transaction on Wireless Communications*, 8(3), 2009.
- [Bil07] A. BILLIONNET : *Optimisation discrète*. Dunod, 2007.
- [Bir49] G. BIRKHOFF : Théorie et applications des treillis. *Annales de l'institut Henri Poincaré*, 11(5):227–240, 1949.
- [Bir67] G. BIRKHOFF : *Lattice theory*, volume 25. AMS Colloquium Publications, 3ème édition, 1967.
- [BJI08] H. M. N. D. BANDARA, A. P. JAYASUMANA et T. H. ILLANGASEKARE : Cluster tree based self organization of virtual sensor networks. Dans *Wireless Mesh and Sensor Networks*, 2008.
- [BYAH06] M. BUETTNER, G. V. YEE, E. ANDERSON et R. HAN : X-MAC : A short preamble MAC protocol for duty-cycled wireless sensor networks. Dans *SenSys (ACM Conference on Embedded Networked Sensor Systems)*, pages 307–320, 2006.
- [CC1020] CC1020 - Low-power RF transceiver for narrowband systems. Datasheet revision 1.5, Chipcon AS, 2004.
- [CC2420] CC2420 - 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF transceiver. Preliminary datasheet revision 1.2, Chipcon, 2004.
- [CGJ+08*] G. CHALHOUB, A. GUITTON, F. JACQUET, A. FREITAS et M. MISSON : Medium access control for a tree-based wireless sensor network: Synchronization management. Dans *IFIP Wireless Days*, novembre 2008.
- [CGM08*] G. CHALHOUB, A. GUITTON et M. MISSON : MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part A: MaCARI: a synchronized tree-based MAC protocol. Dans *WSAN (IFIP Conference on Wireless Sensor and Actor Networks)*, juillet 2008.
- [Cha09] G. CHALHOUB : *MaCARI : une méthode d'accès déterministe et économe en énergie pour les réseaux de capteurs sans fil*. Thèse de doctorat, Université Blaise Pascal, décembre 2009.

- [CHGM09*] G. CHALHOUB, N. HADID, A. GUITTON et M. MISSON : Deference mechanisms significantly increase the MAC delay of slotted CSMA/CA. Dans *ICC (IEEE International Conference on Communications)*, juin 2009. Taux d'acceptation : 35%.
- [Cle13] CLERMONT UNIVERSITÉ : Clervolc. <http://clervolc.univ-bpclermont.fr/>, septembre 2013.
- [CLG⁺09*] G. CHALHOUB, E. LIVOLANT, A. GUITTON, A. VAN DEN BOSSCHE, M. MISSON et T. VAL : Specifications and evaluation of a MAC protocol for a LP-WPAN. *AHSWN (Ad Hoc & Sensor Wireless Networks – An International Journal)*, 7 (1):69–89, 2009.
- [CMP⁺13] F. CLAD, P. MERINDOL, J.-J. PANSIOT, P. FRANÇOIS et O. BONAVENTURE : Graceful convergence in link-state IP networks: a lightweight algorithm ensuring minimal operational impact. *IEEE/ACM Transactions on Networking*, 2013.
- [CSL09] J. CHILO, A. SCHLUTER et T. LINDBLAD : *Development of a High-Resolution Wireless Sensor Network for Monitoring Volcanic Activity*, chapitre 3. InTech, 2009.
- [CSS02] D. CAVIN, Y. SASSON et A. SCHIPER : On the accuracy of MANET simulators. Dans *POMC (Principles of Mobile Computing)*, pages 38–43, 2002.
- [Cza06] P. P. CZAPSKI : A survey: MAC protocols for applications of wireless sensor networks. Dans *TENCON (IEEE Technical Conference of IEEE Region 10)*, pages 1–4, 2006.
- [DD08] T. DANG et C. DEVIC : OCARI: optimization of communication for ad hoc reliable industrial networks. Dans *INDIN (IEEE International Conference on Industrial Informatics)*, 2008.
- [DEA06] I. DEMIRKOL, C. ERSOY et F. ALAGOZ : MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, 2006.
- [DEM⁺09] V. DYO, S. A. ELLWOOD, D. W. MACDONALD, A. MARKHAM, C. MASCOLO, B. PÁSZTOR, N. TRIGONI et R. WOHLERS : Wildlife and environmental monitoring using RFID and WSN technology. Dans *SenSys (ACM Conference on Embedded Networked Sensor Systems)*, pages 371–372, 2009.
- [DGMD11*] F. DELOBEL, A. GUITTON, M. MISSON et W. DARGIE : Minimization of the diffusion delay of a tree-based wireless sensor network. Dans *Globecom (IEEE Global Communications Conference)*, 2011. Taux d'acceptation : 37%.
- [DL05] M. DITZEL et K. LANGENDOEN : D3: data-centric data dissemination in wireless sensor networks. Dans *European Conference on Wireless Technology*, pages 185–188, 2005.
- [DLPF09] F. DIDI, H. LABIOD, G. PUJOLLE et M. FEHAM : Mobility and QoS of 802.11 and 802.11e wireless LAN standards. *International Arab Journal of Information Technology*, 6(4):403–411, 2009.
- [EEHDP04] C. C. ENZ, A. EL-HOIYDI, J.-D. DECOTIGNIE et V. PEIRIS : WiseNET: an ultralow-power wireless sensor network solution. *IEEE Computers*, 37(8):62–70, 2004.

- [EGE02] J. ELSON, L. GIROD et D. ESTRIN : Fine-grained network time synchronization using reference broadcasts. Dans *OSDI (USENIX Symposium on Operating Systems Design and Implementation)*, décembre 2002.
- [ER11] N. EL RACHKIDY : *Cross-layering et routage dans un réseau ad hoc : politique de relais de trame sur un réseau de capteurs sans fil organisé selon une topologie en arbre*. Thèse de doctorat, Université Blaise Pascal, décembre 2011.
- [ERCGM11*] N. EL RACHKIDY, G. CHALHOUB, A. GUITTON et M. MISSON : Queue-exchange mechanism to improve the QoS in a multi-stack architecture. Dans *PE-WASUN (ACM International Symposium on Performance Evaluation of Wireless Ad hoc, Sensor and Ubiquitous Networks)*, 2011.
- [ERGB13*] N. EL RACHKIDY, A. GUITTON et C. BURATTI : Improving the AODV-based ZigBee routing protocol through pivots. Dans *PIMRC (IEEE International Symposium on Personal, Indoor and Mobile Radio Communications)*, 2013.
- [ERGBM09*] N. EL RACHKIDY, A. GUITTON, B. BAKHACHE et M. MISSON : PiRAT: Pivot routing for alarm transmission in wireless sensor networks. Dans *LCN (IEEE Local Computer Networks)*, octobre 2009. Taux d'acceptation : 31%.
- [ERGBM10*] N. EL RACHKIDY, A. GUITTON, B. BAKHACHE et M. MISSON : Address assignment for wireless sensor networks in mines. Dans *ICWCUCA (International Conference on Wireless Communications in Underground and Confined Areas)*, août 2010. Conférence reportée à août 2012.
- [ERGM09*] N. EL RACHKIDY, A. GUITTON et M. MISSON : Optimizing the setup phase of an IEEE 802.15.4 wireless sensor network. Dans *IFIP Wireless Days*, décembre 2009.
- [ERGM10*] N. EL RACHKIDY, A. GUITTON et M. MISSON : Routing protocol for anycast communications in a wireless sensor network. Dans *Networking (IFIP Networking Conference)*, LNCS 6091, pages 291–302, avril 2010. Taux d'acceptation : 24%.
- [ERGM11*] N. EL RACHKIDY, A. GUITTON et M. MISSON : Improving QoS in wireless sensor networks using a multi-stack architecture. Dans *VTC (IEEE Vehicular Technology Conference)*, mai 2011. Taux d'acceptation : 51%.
- [ERGM12a*] N. EL RACHKIDY, A. GUITTON et M. MISSON : *Introduction to routing issues*, chapitre intitulé Joint sink and route selection for anycast communications in wireless sensor networks. iConcept Press, 2012.
- [ERGM12b*] N. EL RACHKIDY, A. GUITTON et M. MISSON : Pivot routing improves wireless sensor networks performance. *JNW (Journal of Networks)*, 2012.
- [ERGM12c*] N. EL RACHKIDY, A. GUITTON et M. MISSON : *Wireless autonomous devices*, chapitre intitulé Reduction of the association phase of a ZigBee wireless sensor network. Academy Publish, 2012.
- [ERGM13a*] N. EL RACHKIDY, A. GUITTON et M. MISSON : Avoiding routing loops in a multi-stack WSN. *JCM (Journal of Communications)*, 8(3), 2013.
- [ERGM13b*] N. EL RACHKIDY, A. GUITTON et M. MISSON : Improving routing performance when several routing protocols are used sequentially in a WSN. Dans *ICC (IEEE International Conference on Communications)*, 2013. Taux d'acceptation : 39%.

- [FB07] P. FRANÇOIS et O. BONAVENTURE : Avoiding transient loops during the convergence of link-state routing protocols. *IEEE/ACM Transactions on Networking*, 15(6):1280–1292, 2007.
- [FCA10] M. FONOAGE, M. CARDEI et A. AMBROSE : A QoS based routing protocol for wireless sensor networks. Dans *IPCCC (IEEE Performance Computing and Communications Conference)*, 2010.
- [FreeScale] ZigBee reference design. Rapport technique ZRD01, FreeScale Semiconductor, 2004.
- [FvdBV12] N. FOURTY, A. van den BOSSCHE et T. VAL : An advanced study of energy consumption in an ieee 802.15.4 based network : everything but the truth on 802.15.4 node lifetime. *Computer Communications*, 35:1759–1767, 2012.
- [GBO⁺10] D. L. GUIDONI, A. BOUKERCHE, H. A. B. F. OLIVEIRA, R. A. F. MINI et A. A. F. LOUREIRO : A small world model to improve synchronization algorithms for wireless sensor networks. Dans *ISCC (IEEE Symposium on Computers and Communications)*, 2010.
- [GH09*] A. GUITTON et N. HADID : Analysis of channel access delay in slotted CSMA/CA in a WSN. Dans *S-Cube (International Conference on Sensor Systems and Software)*, LNICST 24, pages 83–97, septembre 2009. Papier invité.
- [GKS03] S. GANERIWAL, R. KUMAR et M. B. SRIBASTAVA : Timing-sync protocol for sensor networks. Dans *SenSys (ACM Embedded Networked Sensor Systems)*, pages 138–149, 2003.
- [GR13*] A. GUITTON et O. RAYNAUD : Treillis et réseaux de capteurs sans fil. Travail en cours, 2013.
- [GSGSL⁺10] A.-J. GARCIA-SANCHEZ, F. GARCIA-SANCHEZ, F. LOSILLA, P. KULAKOWSKI, J. GARCIA-HARO, A. RODRÍGUEZ, J.-V. LÓPEZ-BAO et F. PALOMARES : Wireless sensor network deployment for monitoring wildlife passages. *Sensors*, 10:7236–7262, 2010.
- [GTH08*] A. GUITTON, N. TRIGONI et S. HELMER : Fault-tolerant compression algorithms for sensor networks with unreliable links. Dans *DCOSS (IEEE International Conference on Distributed Computing in Sensor Systems)*, pages 190–203, juin 2008.
- [Had11] N. HADID : *Utilisation d’un canal d’aparté pour des applications de réseaux de capteurs embarquées*. Thèse de doctorat, Université Blaise Pascal, juillet 2011.
- [Hak71] S. L. HAKIMI : Steiner’s problem in graphs and its implications. *Networks*, 1:113–133, 1971.
- [HB07] M. HEFEEDA et M. BAGHERI : Wireless sensor networks for early detection of forest fires. Dans *MASS (IEEE Mobile Adhoc and Sensor Systems)*, pages 1–6, 2007.
- [HC12] L. HUANG et S. CHENG : Unmanned monitoring system of rivers and lakes based on WSN. Dans *ICSAI (International Conference on Systems and Informatics)*, pages 495–498, 2012.
- [HGM09a*] N. HADID, A. GUITTON et M. MISSON : Adaptation de l’algorithme de *backoff* de CSMA/CA slotté au trafic cumulé. Dans *JDIR (Journées Doctorales en Informatique et Réseau)*, février 2009.

- [HGM09b*] N. HADID, A. GUITTON et M. MISSON : Adaptive slotted CSMA/CA algorithm for the traffic accumulated during the inactive period. Dans *PE-WASUN (ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks)*, octobre 2009. Papier court.
- [HGM10*] N. HADID, A. GUITTON et M. MISSON : Exploiting a meeting channel to interconnect 802.15.4-compliant mobile entities: discovery and association phases. Dans *ISCC (IEEE Symposium on Computers and Communications)*, juin 2010. Taux d'acceptation : 40%.
- [HGM11a*] N. HADID, A. GUITTON et M. MISSON : Exploiting a meeting channel to interconnect mobile robots. *JNCA (Journal of Network and Computer Applications)*, 2011.
- [HGM11b*] N. HADID, A. GUITTON et M. MISSON : Using a meeting channel and relay nodes to interconnect mobile robots. Dans *Mobile Robots / Book 2*. Intech, 2011. ISBN 978-953-307-842-7.
- [HLP11] P. HUNG LE et G. PUJOLLE : A link-disjoint interference-aware multi-path routing protocol for mobile ad hoc network. Dans *DICTAP (Digital Information and Communication Technology and Its Applications)*, volume 1, pages 649–661, 2011.
- [HSB10] T. HARMS, S. SEDIGH et F. BASTIANINI : Structural health monitoring of bridges using wireless sensor networks. *IEEE Instrumentation & Measurement Magazine*, 13(6):14–18, 2010.
- [HUM⁺11] D. HUGHES, J. UEYAMA, E. MENDIONDO, N. MATTHYS, W. HORRÉ, S. MICHIELS, C. HUYGENS, W. JOOSEN, K. L. MAN et S.-U. GUAN : A middleware platform to support river monitoring using wireless sensor networks. *Journal of Brazilian Computer Society*, 17(2):85–102, 2011.
- [HXS⁺13] P. HUANG, L. XIAO, S. SOLTANI, M. W. MUTKA et N. XI : The evolution of MAC protocols in wireless sensor networks: a survey. *IEEE Communications Surveys & Tutorials*, 15(1):101–120, 2013.
- [IEEE 802.11e v2005] IEEE 802.11 : Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications, amendment 8: medium access control (MAC) quality of service enhancements. Standard ieee 802.11e, ANSI/IEEE, 2005.
- [IEEE 802.15-4 v2003] IEEE 802.15 : Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). Standard 802.15.4 R2003, ANSI/IEEE, 2003.
- [IEEE 802.15-4 v2011] IEEE 802.15 : Part 15.4: low-rate wireless personal area networks (LR-WPANs). Standard for local and metropolitan area networks IEEE Std 802.15.4-2011, IEEE, 2011.
- [ITU 1238] ITU : Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900 MHz to 100 GHz. ITU-R P.1238-7, ITU, 2012.
- [ITU 833] ITU : Attenuation in vegetation. ITU-R P.833-6, ITU, 2007.

- [Jav11] P. JAVIER DEL CID : Optimizing resource use in multi-purpose WSNs. Dans *PerCom Workshops (Pervasive Computing Workshops)*, pages 395–396, 2011.
- [JHI07] A. P. JAYASUMANA, Q. HAN et T. H. ILLANGASEKARE : Virtual sensor networks – a resource efficient approach for concurrent applications. Dans *ITNG (International Conference on Information Technology - New Generations)*, 2007.
- [JHMJ10] P. JAVIER DEL CID, D. HUGHES, S. MICHIELS et W. JOOSEN : Expressing and configuring quality of data in multi-purpose wireless sensor networks. Dans *S-Cube (International Conference on Sensor Systems and Software)*, pages 91–106, 2010.
- [JMjH10] P. JAVIER DEL CID, S. MICHIELS, W. JOOSEN et D. HUGHES : Middleware for resource sharing in multi-purpose wireless sensor networks. Dans *NESEA (International Conference on Networked Embedded Systems for Every Application)*, pages 1–8, 2010.
- [JRO10] R. JURDAK, A. G. RUZZELLI et G. M. P. O’HARE : Radio sleep mode optimization in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(7):955–968, 2010.
- [Kar72] R. M. KARP : Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.
- [KKP⁺07] T. KIM, D. KIM, N. PARK, S.-E. YOO et T. S. LÓPEZ : Shortcut tree routing in ZigBee networks. Dans *ISWPC (International Symposium on Wireless and Pervasive Computing)*, 2007.
- [KPC⁺07] S. KIM, S. PAKZAD, D. CULLER, J. DEMMEL, G. FENVES, S. GLASER et M. TURON : Health monitoring of civil infrastructures using wireless sensor networks. Dans *IPSN (ACM International Conference on Information Processing in Sensor Networks)*, pages 254–263, 2007.
- [KRKI03] R. KALIDINDI, L. RAY, R. KANNAN et S. IYENGAR : Distributed energy aware MAC layer protocol for wireless sensor networks. Dans *ICWN (International Conference on Wireless Networks)*, 2003.
- [Kru56] J. KRUSKAL : On the shortest spanning tree of a graph and the traveling salesman problem. *American Mathematical Society*, 7(1):48–50, 1956.
- [KS96] A. KEZDY et S. SEIF : When is a poset isomorphic to the poset of connected induced subgraphs of a graph?. *Southwest Journal of Pure and Applied Mathematics*, 1:42–50, 1996.
- [LGC04] P. LEVIS, D. GAY et D. CULLER : Bridging the gap: programming sensor networks with application specific virtual machines. Tech Report UCB//CSD-04-1343, UC Berkeley, 2004.
- [LTZ⁺13] G. LIU, R. TAN, R. ZHOU, G. XING, W.-Z. SONG et J. M. LEES : Volcanic earthquake timing in wireless sensor networks. Dans *IPSN (ACM/IEEE International Conference on Information Processing in Sensor Networks)*, 2013.
- [LVA07] L. LI, M. C. VURAN et I. F. AKYILDIZ : Characteristics of underground channel for wireless underground sensor networks. Dans *Annual Mediterranean Ad Hoc Networking Workshop*, pages 92–99, 2007.

- [LvdBC11] J. LU, A. van den BOSSCHE et E. CAMPO : A new beacon scheduling mechanism for mesh wireless personal area networks based on IEEE 802.15.4. Dans *ETFA (Emerging Technologies and Factory Automation)*, pages 1–4, 2011.
- [LvdBV08] E. LIVOLANT, A. van den BOSSCHE et T. VAL : MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part B: Optimization of the intra-star exchanges for MaCARI. Dans *WSAN (IFIP Conference on Wireless Sensor and Actor Networks)*, juillet 2008.
- [LXX⁺06] S. LEI, H. XU, W. XIAOLING, Z. LIN, J. CHO et S. LEE : VIP bridge: integrating several sensor networks into one virtual sensor network. Dans *ICISP (International Conference on Internet Surveillance and Protection)*, 2006.
- [LZHL11] C. LI, H. ZHANG, B. HAO et J. LI : A survey on routing protocols for large-scale wireless sensor networks. *Sensors*, 11:3498–3526, 2011.
- [Mah10] S. MAHFOUDH : *Efficacité énergétique dans les réseaux ad hoc et de capteurs : routage, ordonnancement de l'activité des nœuds et cross layering*. Thèse de doctorat, Université Pierre et Marie Curie, 2010.
- [MBWP08] A. MARTIROSYAN, A. BOUKERCHE, R. WERNER et N. PAZZI : A taxonomy of cluster-based routing protocols for wireless sensor networks. Dans *ISpan (International Symposium on Parallel Architectures, Algorithms and Networks)*, pages 247–253, 2008.
- [MCHM13] A. MOUAWAD, G. CHALHOUB, G. HABIB et M. MISSON : A performance study of mobile nodes in a wireless sensor network. Dans *ICCIT (International Conference on Communications and Information Technology)*, 2013. Poster.
- [MCM⁺10] S. MAHFOUDH, G. CHALHOUB, P. MINET, M. MISSON et I. AMDOUNI : Node coloring and color conflict detection in wireless sensor networks. *Future Internet*, 2(4):469–504, 2010.
- [McN96] S. R. MCNUTT : Seismic monitoring and eruption forecasting of volcanoes: a review of the state of the art and case histories. Dans SCARPA et TILLING, éditeurs : *Monitoring and Mitigation of Volcano Hazards*, pages 99–146. Springer-Verlag Berlin Heidelberg, 1996.
- [Mem13] MEMSIC : TelosB. http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf, septembre 2013.
- [MFHH05] S. R. MADDEN, M. J. FRANKLIN, J. M. HELLERSTEIN et W. HONG : TinyDB: an acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems (TODS)*, 30(1):122–173, 2005. Special Issue on SIGMOD/PODS 2003.
- [MG07] D. MANJUNATH et S. V. GOPALIAH : A multi-purpose wireless sensor network for residential layouts. Dans *BWCCA (International Conference on Broadband and Wireless Computing, Communication and Applications)*, pages 49–58, 2007.
- [MHJ⁺12] S. MOAD, M. T. HANSEN, R. JURDAK, B. KUSY, N. BOUABDALLAH et A. KSENTINI : On balancing between minimum energy and minimum delay with radio diversity for wireless sensor networks. Dans *IFIP Wireless Days*, 2012.

- [MKSL04] M. MARÓTI, B. KUSY, G. SIMON et A. LÉDECZI : The flooding time synchronization protocol. Dans *SenSys (ACM Conference on Embedded Networked Sensor Systems)*, 2004.
- [MM08a] S. MAHFOUDH et P. MINET : EOLSR: an energy efficient routing protocol in wireless ad hoc and sensor networks. *JOIN (Journal of Interconnection Networks)*, 2008. Numéro spécial sur Routing, Scheduling & Load balancing in Networking Systems.
- [MM08b] P. MINET et S. MAHFOUDH : SERENA: scheduling router nodes activity in wireless ad hoc and sensor networks. Dans *IWCMC (IEEE International Wireless Communications and Mobile Computing Conference)*, août 2008.
- [MMCG10*] P. MINET, S. MAHFOUDH, G. CHALHOUB et A. GUITTON : Node coloring in a wireless sensor network with unidirectional links and topology changes. Dans *WCNC (IEEE Wireless Communications and Networking Conference)*, avril 2010. Taux d'acceptation : 37%.
- [MMKR13] J. MISIC, V. B. MISIC, M. S. I. KHAN et M. M. RAHMAN : Properties of blind rendezvous in channel hopping cognitive piconets. Dans *VTC (IEEE Vehicular Technology Conference)*, 2013.
- [NS04] P. NAIK et K. M. SIVALINGAM : *Wireless sensor networks*, chapitre intitulé A survey of MAC protocols for sensor networks, pages 93–107. Springer, 2004.
- [OSI] Information technology - open systems interconnection - basic reference model: The basic model. International standard 7498-1, ISO/IEC, 1994.
- [PHC04] J. POLASTRE, J. HILL et D. CULLER : Versatile low power media access for wireless sensor networks. Dans *SenSys (ACM Conference on Embedded Networked Sensor Systems)*, pages 95–107, 2004.
- [PKSH07] K. J. PAEK, J. KIM, U. S. SONG et C. S. HWANG : Priority-based medium access control protocol for providing QoS in wireless sensor networks. *IEICE Transactions on Information and Systems*, pages 1448–1451, 2007.
- [Pri57] R. C. PRIM : Shortest connection networks and some generalizations. *The Bell Systems Technical Journal*, 36(11):1389–1401, 1957.
- [PTBD10] B. PAVKOVIC, F. THEOLEYRE, D. BARTHEL et A. DUDA : Experimental analysis and characterization of a wireless sensor network environment. Dans *PEWASUN (ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks)*, 2010.
- [Rez12] A. REZZAG : Déploiement d'un réseau de capteurs sans fil, partiellement enterré et en extérieur. Mémoire de M2 recherche, Université Blaise Pascal, 2012.
- [RFC 3493] R. GILLIGAN, S. THOMSON, J. BOUND, J. McCANN et W. STEVENS : Basic socket interface extensions for IPv6. RFC 3493, IETF, 2003.
- [RFC 3561] C. PERKINS, E. BELDING-ROYER et S. DAS : Ad hoc on-demand distance vector (AODV) routing. RFC 3561, IETF, 2003.
- [RFC 3626] C. ADJIH, T. CLAUSEN, P. JACQUET, A. LAOUITI, P. MINET, P. MUELHALER, A. QAYYUM et L. VIENNOT : Optimized link state routing protocol. RFC 3626, IETF, 2003.

- [RFC 4213] E. NORDMARK et R. GILLIGAN : Basic transition mechanisms for IPv6 hosts and routers. RFC 4213, IETF, 2005.
- [RGB⁺11] H. S. RAMOS, D. L. GUIDONI, A. BOUKERCHE, E. F. NAKAMURA, A. C. FRERY et A. A. F. LOUREIRO : Topology-related modeling and characterization of wireless sensor networks. Dans *PE-WASUN (ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks)*, pages 33–40, 2011.
- [ROGLA03] V. RAJENDRAN, K. OBRACZKA et J. J. GARCIA-LUNA-ACEVES : Energy-efficient, collision-free medium access control for wireless sensor networks. Dans *SenSys (ACM Conference on Embedded Networked Sensor Systems)*, pages 181–192, 2003.
- [RPA⁺13] G. ROMANIELLO, E. POTETSIAKIS, O. ALPHAND, R. GUIZZETTI et A. DUDA : Fast and energy-efficient topology construction in multi-hop multi-channel 802.15.4 networks. Dans *WIMOB (IEEE Conference on Wireless and Mobile Computing, Networking and Communications)*, 2013.
- [SFCB05] J. STEFFAN, L. FIEGE, M. CILIA et A. P. BUCHMANN : Towards multi-purpose wireless sensor networks. Dans *Systems Communications*, pages 336–341, 2005. Session SENET.
- [SHC⁺08] L. SHU, M. HAUSWIRTH, L. CHENG, J. MA, V. REYNOLDS et L. ZHANG : Sharing worldwide sensor network. Dans *SAINT (International Symposium on Applications and the Internet)*, pages 189–192, 2008.
- [SHX⁺09] W.-Z. SONG, R. HUANG, M. XU, A. MA, B. SHIRAZI et R. LAHUSEN : Air-dropped sensor network for real-time high-fidelity volcano monitoring. Dans *MobiSys (ACM International Conference on Mobile Systems)*, 2009.
- [SPM06] E. P. STUNTEBECK, D. POMPILI et T. MELODIA : Wireless underground sensor networks using commodity terrestrial motes. Dans *SECON (IEEE International Conference on Sensing, Communication, and Networking)*, pages 112–114, 2006.
- [SSDQ09] R. S. SUDHAAKAR, A. SANZGIRI, M. DEMIRBAS et C. QIAN : A plant-and-play wireless sensor network system for gate monitoring. Dans *WOWMOM (International Symposium on a World of Wireless Mobile and Multimedia Networks)*, pages 1–9, 2009.
- [SSHK⁺12] R. SHIRANI, M. ST-HILAIRE, T. KUNZ, Y. ZHOU, J. LI et L. LAMONT : Combined reactive-geographic routing for unmanned aeronautical adhoc networks. Dans *IWCMC (International Wireless Communications and Mobile Computing Conference)*, 2012.
- [SSJZ08] I. SLAMA, B. SHRESTHA, B. JOUABER et D. ZEGHLACHE : A hybrid MAC with prioritization for wireless sensor networks. Dans *LCN (IEEE Conference on Local Computer Networks)*, pages 274–281, 2008.
- [Ste90] J. H. STEELMAN : The lattice of connected subgraphs of a graph, solution to Problem E3281. *American Mathematical Monthly*, 97:924–925, 1990.
- [Sud10] M. SUDHEER : *Wireless sensor networks: Application-centric design*, chapitre intitulé Wireless sensor network for disaster monitoring. InTech, 2010.

- [SV05] J. STEFFAN et M. VOSS : Security mechanisms for multi-purpose sensor networks. Dans *GI Jahrestagung*, volume 2, pages 158–160, 2005.
- [SV10] A. R. SILVA et M. C. VURAN : Communication with aboveground devices in wireless underground sensor networks: an empirical study. Dans *ICC (IEEE International Conference on Communications)*, pages 1–6, 2010.
- [SZLD12] L. SARAKIS, T. ZAHARIADIS, H.-C. LELIGOU et M. DOHLER : A framework for service provisioning in virtual sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2012.
- [TBZ⁺05] D. M. TRALLI, R. G. BLOM, V. ZLOTNICKI, A. DONNELLAN et D. L. EVANS : Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(4):185–198, 2005.
- [TLZ⁺13] P. TRAKADAS, H.-C. LELIGOU, T. ZAHARIADIS, P. KARKAZIS et L. SARAKIS : Managing QoS for future internet applications over virtual sensor networks. Dans *FIA (Future Internet Assembly)*, pages 52–63, 2013.
- [TOOM08] R. TYNAN, G. M. P. O’HARE, M. J. O’GRADY et C. MULDOON : Virtual sensor networks: an embedded agent. Dans *ISPA (International Symposium on Parallel and Distributed Processing with Applications)*, 2008.
- [TXC⁺10] R. TAN, G. XING, J. CHEN, W.-Z. SONG et R. HUANG : Quality-driven volcanic earthquake detection using wireless sensor networks. Dans *RTSS (IEEE Real-Time Systems Symposium)*, pages 271–280, 2010.
- [Uni13] UNIVERSITY OF CALIFORNIA, BERKELEY : TinyOS. <http://www.tinyos.net>, septembre 2013.
- [Var09] G. VARAPRASAD : Wireless sensor networks for volcano environments. *ACM SIGBED Review, special issue on adaptive and reconfigurable embedded systems*, 6(3), 2009.
- [vdB07] A. van den BOSSCHE : *Proposition d’une nouvelle méthode d’accès déterministe pour un réseau personnel sans fil à fortes contraintes temporelles*. Thèse de doctorat, Université de Toulouse, juillet 2007.
- [vdBVD11] A. van den BOSSCHE, T. VAL et R. DALCÉ : SISP: a lightweight synchronization protocol for wireless sensor networks. Dans *ETFA (Emerging Technologies and Factory Automation)*, 2011.
- [VOCA09] L. J. G. VILLALBA, A. L. S. OROZCO, A. T. CABRERA et C. J. B. ABBAS : Routing protocols in wireless sensor networks. *Sensors*, 9:8399–8421, 2009.
- [WAJR⁺05] G. WERNER-ALLEN, J. JOHNSON, M. RUIZ, J. LEES et M. WELSH : Monitoring volcanic eruptions with a wireless sensor network. Dans *EWSN (European Workshop on Sensor Networks)*, pages 108–120, 2005.
- [WALJ⁺06] G. WERNER-ALLEN, K. LORINCZ, J. JOHNSON, J. LEES et M. WELSH : Fidelity and yield in a volcano monitoring sensor network. Dans *OSDI (USENIX Symposium on Operating Systems Design and Implementation)*, pages 381–396, 2006.
- [WALW⁺06] G. WERNER-ALLEN, K. LORINCZ, M. WELSH, O. MARCILLO, J. JOHNSON, M. RUIZ et J. LEES : Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

- [WMG04] A. WOO, S. MADDEN et R. GOVINDAN : Networking support for query processing in sensor networks. *Communications of the ACM*, 6(47):47–52, 2004. Special issue on Wireless sensor networks.
- [WTC03] A. WOO, T. TONG et D. CULLER : Taming the underlying challenges of reliable multihop routing in sensor networks. Dans *SenSys (ACM Conference on Embedded Networked Sensor Systems)*, pages 14–27, 2003.
- [YHE04] W. YE, J. HEIDEMANN et D. ESTRIN : Medium access control with coordinated adaptive sensing for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.
- [YIE11] M. A. YIGITEL, O. D. INCEL et C. ERSOY : QoS-aware MAC protocols for wireless sensor networks: a survey. *Computer Networks*, 55(8):1982–2004, 2011.
- [YM10] Z. YANG et A. MOHAMMED : *Sustainable wireless sensor networks*, chapitre intitulé A survey on routing protocols for wireless sensor networks. Intech, 2010.
- [YVM09] R. YADAV, S. VARMA et N. MALAVIYA : A survey of MAC protocols for wireless sensor networks. *UbiCC*, 4(3):827–833, 2009.
- [YWM05] L. YU, N. WANG et X. MENG : Real-time forest fire detection with wireless sensor networks. Dans *International Conference on Wireless Communications, Networking and Mobile Computing*, volume 2, pages 1214–1217, 2005.
- [Zha05] Y. ZHANG : Wireless sensor network for volcano monitoring. Mémoire de M2 recherche, Royal Institute of Technology (KTH), Sweden, 2005.
- [ZigBee] ZIGBEE STANDARDS ORGANIZATION : ZigBee Specification. Document 053474r17, ZigBee Standards Organization, 2008.
- [Zim80] H. ZIMMERMANN : OSI reference model - the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4): 425–423, 1980.

Deuxième partie

Synthèse de mes activités d'enseignant-chercheur

CHAPITRE

6

PARCOURS

J'ai effectué ma thèse de doctorat en informatique à l'IRISA (Institut de Recherche en Informatique et en Systèmes Aléatoires) / Université de Rennes I, de septembre 2002 à octobre 2005, dans l'équipe ARMOR (Architectures et Modèles Réseaux), sous la direction du Pr. Raymond MARIE et du Dr Miklós MOLNÁR. Ma thèse est intitulée : « Communications *multicast* : contributions au passage à l'échelle et aux réseaux tout optique ». Elle traite de l'optimisation des communications de groupes dans les réseaux filaires, au niveau de la couche réseau. Je l'ai soutenue le 7 octobre 2005, devant le jury constitué de Pr. Éric FLEURY (rapporteur), Pr. Gérard HÉBUTERNE (président), Pr. Jean-Claude KÖNIG (rapporteur), Pr. Raymond MARIE (directeur) et Dr Miklós MOLNÁR (co-directeur).

J'ai ensuite effectué deux post-doctorats sous l'encadrement du Dr Niki TRIGONI et dans l'équipe ASSENT (*Ad-hoc Scalable Sensor Network*), l'un à *Birkbeck College, University of London*, d'octobre 2005 à mars 2007, et l'autre à *Oxford University Computing Laboratory, University of Oxford*, d'avril 2007 à août 2007. J'ai travaillé sur les réseaux de capteurs sans fil, au niveau de la couche réseau et de la couche application.

Depuis septembre 2007, je suis maître de conférences à l'UFR Sciences et Technologies de l'Université Blaise Pascal. Je fais ma recherche dans l'équipe « réseaux et protocoles », dirigée par le Pr. Michel MISSON. Cette équipe fait partie de l'axe 2 (systèmes d'information et de communication) du laboratoire LIMOS (Laboratoire d'Informatique, de Modélisation, et d'Optimisation des Systèmes). Ma recherche se focalise sur les réseaux de capteurs sans fil, et principalement sur la conception, la spécification, la simulation et le prototypage de protocoles, au niveau des couches liaison de données et réseau.

CHAPITRE

7

PRODUCTION SCIENTIFIQUE

Dans ce chapitre, je fais une synthèse de ma production scientifique en termes de papiers acceptés et d'indicateurs bibliométriques.

7.1 Tableau récapitulatif

Dans le tableau récapitulatif suivant :

- seuls figurent les revues, chapitres de livres et conférences avec comité de relecture,
- les conférences internationales sont regroupées selon leur classement (référence : classement australien CORE¹ de décembre 2007),
- les revues et conférences qui ne sont pas organisées par une institution scientifique renommée (ACM, IEEE ou IFIP) et dont les actes ne sont pas publiés dans des séries reconnues (LNCS ou LNICST) apparaissent entre parenthèses,
- les conférences francophones sont considérées comme nationales, ainsi que les conférences anglophones impliquant un nombre restreint de pays.

Année	Revue		Chap. de livre	Conférences internationales			Conf. nat.	Total
	int.	nat.		papier long	papier court	groupe de travail		
2013	1			1A + 1B (+1)			(1)	5
2012	1		2					3
2011	1		1	2B + 1				5

1. Computing Research and Education Association of Australasia : Conference Rankings, <http://www.core.edu.au/>, septembre 2013.

Année	Revue		Chap. de livre	Conférences internationales			Conf. nat.	Total
	int.	nat.		papier long	papier court	groupe de travail		
2010				1A + 2B (+1)				4
2009	1 (+2)			2A + 2	1		(1)	9
2008				2B + 3				5
2007			1	1B				2
2006	(1)			1A + 1B	1A ⁺ + 1	(1)		6
2005				1 (+1)	1A		(1)	4
2004		(1)		1B + 1				3
2003				1 (+2)			(4)	7
Total	7	1	4	29	4	1	7	53

7.2 Indicateurs bibliométriques

Les indicateurs bibliométriques permettent une analyse quantitative de l'activité scientifique au travers des publications. Toutefois, ces indicateurs sont sujets à de nombreuses controverses en raison de nombreux biais, comme le précise notamment un rapport de Inria², et doivent donc être considérés avec prudence.

Indicateurs bibliométriques selon *google scholar* (au 28 janvier 2014) :

- citations : 264 depuis 2009, 382 au total,
- h-index³ : 9 depuis 2009, 12 au total,
- i10-index⁴ : 9 depuis 2009, 14 au total,
- a-index⁵ (calculé manuellement avec les résultats de *google scholar*) : 19.83 au total,
- g-index⁶ (calculé manuellement avec les résultats de *google scholar*) : 16 au total.

Nombre moyen d'auteurs par papier :

- revues (internationales et nationales) : 4,75,
- chapitres de livres : 3,
- conférences internationales : 3,40.

Nombre total de co-auteurs :

- revues (internationales et nationales) : 18,
- chapitres de livres : 5,

2. A.-M. KERMARREC, E. FAOU, J.-P. MERLET (rapporteur), P. ROBERT, L. SEGOUFIN : Que mesurent les indicateurs bibliométriques ? Document d'analyse de la Commission d'évaluation de INRIA, validé par la Commission d'Évaluation du 12 septembre 2007.

3. Un h-index de x indique que x publications ont été citées au moins x fois.

4. Un i10-index de x indique que x publications ont été citées au moins 10 fois.

5. Le a-index correspond au nombre moyen de citations pour les articles retenus dans le calcul du h-index.

6. Un g-index de x indique que x articles ont été cités globalement au moins x^2 fois.

- conférences internationales : 32,
- total : 42.

7.3 Liste complète

La liste ci-dessous représente l'ensemble de mes publications (hors manuscrit de thèse) acceptées après avis d'un comité de relecture.

- [1] A. T. ABY, A. GUITTON et M. MISSON : MAC mechanism for a scalable wireless sensor network using independent duty cycles. Dans *NICST (International France-China Workshop on New and Smart Information Communication Science and Technology to Support Sustainable Development)*, 2013.
- [2] N. EL RACHKIDY, A. GUITTON et C. BURATTI : Improving the AODV-based ZigBee routing protocol through pivots. Dans *PIMRC (IEEE International Symposium on Personal, Indoor and Mobile Radio Communications)*, 2013.
- [3] D. ABRIGNANI, C. BURATTI, D. DARDARI, N. EL RACHKIDY, A. GUITTON, F. MARTELLI, A. STAJKIC et R. VERDONE : The EuWIn testbed for 802.15.4/ZigBee networks: From the simulation to the real-world. Dans *ISWCS (International Symposium on Wireless Communication Systems)*, 2013.
- [4] N. EL RACHKIDY, A. GUITTON et M. MISSON : Avoiding routing loops in a multi-stack WSN. *JCM (Journal of Communications)*, 8(3), 2013.
- [5] N. EL RACHKIDY, A. GUITTON et M. MISSON : Improving routing performance when several routing protocols are used sequentially in a WSN. Dans *ICC (IEEE International Conference on Communications)*, 2013. Taux d'acceptation : 39%.
- [6] N. EL RACHKIDY, A. GUITTON et M. MISSON : *Wireless autonomous devices*, chapitre intitulé Reduction of the association phase of a ZigBee wireless sensor network. Academy Publish, 2012.
- [7] N. EL RACHKIDY, A. GUITTON et M. MISSON : Pivot routing improves wireless sensor networks performance. *JNW (Journal of Networks)*, 2012.
- [8] N. EL RACHKIDY, A. GUITTON et M. MISSON : *Introduction to routing issues*, chapitre intitulé Joint sink and route selection for anycast communications in wireless sensor networks. iConcept Press, 2012.
- [9] N. HADID, A. GUITTON et M. MISSON : Exploiting a meeting channel to interconnect mobile robots. *JNCA (Journal of Network and Computer Applications)*, 2011.
- [10] N. EL RACHKIDY, G. CHALHOUB, A. GUITTON et M. MISSON : Queue-exchange mechanism to improve the QoS in a multi-stack architecture. Dans *PE-WASUN (ACM International Symposium on Performance Evaluation of Wireless Ad hoc, Sensor and Ubiquitous Networks)*, 2011. Taux d'acceptation : 38%. Prix du meilleur papier.
- [11] F. DELOBEL, A. GUITTON, M. MISSON et W. DARGIE : Minimization of the diffusion delay of a tree-based wireless sensor network. Dans *Globecom (IEEE Global Communications Conference)*, 2011. Taux d'acceptation : 37%.

- [12] N. HADID, A. GUITTON et M. MISSON : Using a meeting channel and relay nodes to interconnect mobile robots. Dans *Mobile Robots / Book 2*. Intech, 2011. ISBN 978-953-307-842-7.
- [13] N. EL RACHKIDY, A. GUITTON et M. MISSON : Improving QoS in wireless sensor networks using a multi-stack architecture. Dans *VTC (IEEE Vehicular Technology Conference)*, mai 2011. Taux d'acceptation : 51%.
- [14] N. EL RACHKIDY, A. GUITTON, B. BAKHACHE et M. MISSON : Address assignment for wireless sensor networks in mines. Dans *ICWCUCA (International Conference on Wireless Communications in Underground and Confined Areas)*, août 2010. Conférence reportée à août 2012.
- [15] N. HADID, A. GUITTON et M. MISSON : Exploiting a meeting channel to interconnect 802.15.4-compliant mobile entities: discovery and association phases. Dans *ISCC (IEEE Symposium on Computers and Communications)*, juin 2010. Taux d'acceptation : 40%.
- [16] N. EL RACHKIDY, A. GUITTON et M. MISSON : Routing protocol for anycast communications in a wireless sensor network. Dans *Networking (IFIP Networking Conference)*, LNCS 6091, pages 291–302, avril 2010. Taux d'acceptation : 24%.
- [17] P. MINET, S. MAHFOUDH, G. CHALHOUB et A. GUITTON : Node coloring in a wireless sensor network with unidirectional links and topology changes. Dans *WCNC (IEEE Wireless Communications and Networking Conference)*, avril 2010. Taux d'acceptation : 37%.
- [18] N. EL RACHKIDY, A. GUITTON et M. MISSON : Optimizing the setup phase of an IEEE 802.15.4 wireless sensor network. Dans *IFIP Wireless Days*, décembre 2009. Taux d'acceptation : 38%.
- [19] N. HADID, A. GUITTON et M. MISSON : Adaptive slotted CSMA/CA algorithm for the traffic accumulated during the inactive period. Dans *PE-WASUN (ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks)*, octobre 2009. Papier court.
- [20] N. EL RACHKIDY, A. GUITTON, B. BAKHACHE et M. MISSON : PiRAT: Pivot routing for alarm transmission in wireless sensor networks. Dans *LCN (IEEE Local Computer Networks)*, octobre 2009. Taux d'acceptation : 31%.
- [21] A. GUITTON et N. HADID : Analysis of channel access delay in slotted CSMA/CA in a WSN. Dans *S-Cube (International Conference on Sensor Systems and Software)*, LNICST 24, pages 83–97, septembre 2009. Papier invité.
- [22] K. AL AGHA, M.-H. BERTIN, T. DANG, A. GUITTON, P. MINET, T. VAL et J.-B. VIOLLET : Which wireless technology for industrial wireless sensor networks? the development of Ocari technology. *IEEE Transactions on Industrial Electronics*, 56(10), 2009.
- [23] K. AL AGHA, G. CHALHOUB, A. GUITTON, E. LIVOLANT, S. MAHFOUDH, P. MINET, M. MISSON, J. RAHMÉ, T. VAL et A. VAN DEN BOSSCHE : Cross layering in an industrial wireless sensor network: Case study of Ocari. *JNW (Journal of Networks)*, 6(4), décembre 2009. Numéro spécial sur Wireless sensor networks: theory and practice.

- [24] G. CHALHOUB, N. HADID, A. GUITTON et M. MISSON : Deference mechanisms significantly increase the MAC delay of slotted CSMA/CA. Dans *ICC (IEEE International Conference on Communications)*, 2009. Taux d'acceptation : 35%.
- [25] N. HADID, A. GUITTON et M. MISSON : Adaptation de l'algorithme de *backoff* de CSMA/CA slotté au trafic cumulé. Dans *JDIR (Journées Doctorales en Informatique et Réseau)*, février 2009.
- [26] G. CHALHOUB, E. LIVOLANT, A. GUITTON, A. VAN DEN BOSSCHE, M. MISSON et T. VAL : Specifications and evaluation of a MAC protocol for a LP-WPAN. *AHSWN (Ad Hoc & Sensor Wireless Networks – An International Journal)*, 7 (1):69–89, 2009.
- [27] G. CHALHOUB, A. GUITTON, F. JACQUET, A. FREITAS et M. MISSON : Medium access control for a tree-based wireless sensor network: Synchronization management. Dans *IFIP Wireless Days*, novembre 2008. Taux d'acceptation : 50%.
- [28] G. CHALHOUB, A. GUITTON et M. MISSON : MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part A: MaCARI: a synchronized tree-based MAC protocol. Dans *WSAN (IFIP Conference on Wireless Sensor and Actor Networks)*, juillet 2008.
- [29] A. GUITTON, N. TRIGONI et S. HELMER : Fault-tolerant compression algorithms for sensor networks with unreliable links. Dans *DCOSS (IEEE International Conference on Distributed Computing in Sensor Systems)*, pages 190–203, juin 2008. Taux d'acceptation : 25%.
- [30] J. BACON, A. BERESFORD, D. EVANS, D. INGRAM, N. TRIGONI, A. GUITTON et A. SKORDYLIS : TIME: an open platform for capturing, processing and delivering transport-related data. Dans *CCNC (IEEE Consumer Communications and Networking Conference)*, janvier 2008.
- [31] N. TRIGONI, A. GUITTON et A. SKORDYLIS : Interplay of processing and routing in aggregate query optimization for sensor networks. Dans *ICDCN (International Conference on Distributed Computing and Networking)*, LNCS 4904, pages 401–415, janvier 2008. Taux d'acceptation : 16%.
- [32] A. GUITTON, A. SKORDYLIS et N. TRIGONI : Utilizing correlations to compress time-series in traffic monitoring sensor networks. Dans *WCNC (IEEE Wireless Communications and Networking Conference)*, 2007. Taux d'acceptation : 47%.
- [33] N. TRIGONI, A. GUITTON et A. SKORDYLIS : *Learning from data streams: processing techniques in sensor networks*, chapitre intitulé Querying of sensor data. Springer, 2007.
- [34] A. SKORDYLIS, A. GUITTON et N. TRIGONI : Correlation-based data dissemination in traffic monitoring sensor networks. Dans *CoNext (ACM Conference on Future Networking Technologies)*, page 42, 2006. Papier court.
- [35] J. MOULIERAC, A. GUITTON et M. MOLNÁR : On the number of MPLS LSP using multicast tree aggregation. Dans *Globecom (IEEE Global Communications Conference)*, 2006. Taux d'acceptation : 40%.

- [36] N. TRIGONI, A. GUITTON et A. SKORDYLIS : Poster abstract: Routing and processing multiple aggregate queries in sensor networks. Dans *Sensys (ACM Conference on Embedded Networked Sensor Systems)*, pages 391–392, 2006. Papier court.
- [37] J. MOULIERAC, A. GUITTON et M. MOLNÁR : Hierarchical aggregation of multicast trees in large domains. *JCM (Journal of Communications)*, 1(6):33–44, 2006.
- [38] A. SKORDYLIS, N. TRIGONI et A. GUITTON : A study of approximate data management techniques for sensor networks. Dans *Wises (Workshop on Intelligent Solutions in Embedded Systems)*, 2006. Papier invité.
- [39] J. MOULIERAC, A. GUITTON et M. MOLNÁR : Multicast tree aggregation in large domains. Dans *Networking (IFIP Networking Conference)*, LNCS 3976, pages 691–702, 2006. Taux d’acceptation : 20%.
- [40] A. GUITTON et J. MOULIERAC : Scalable tree aggregation with a large number of multicast groups. Dans *ConTEL (International Conference on Telecommunications)*, 2005. Prix du meilleur papier étudiant.
- [41] J. MOULIERAC et A. GUITTON : QoS scalable tree aggregation. Dans *Networking (IFIP Networking Conference)*, LNCS 3462, pages 1405–1408, 2005. Papier court.
- [42] M. MOLNÁR, A. GUITTON, B. COUSIN et R. MARIE : Efficient protection of many-to-one communications. Dans *EDCC (European Dependable Computing Conference)*, LNCS 3463, pages 363–378, 2005.
- [43] A. GUITTON et A. BOUDANI : Analyse du déploiement incrémental du protocole Xcast. Dans *CFIP (Colloque Francophone sur l’Ingénierie des Protocoles)*, 2005.
- [44] A. BOUDANI, A. GUITTON et B. COUSIN : GXcast: Generalized explicit multicast routing protocol. Dans *ISCC (IEEE Symposium on Computers and Communications)*, pages 1012–1017, 2004. Taux d’acceptation : 39%.
- [45] A. GUITTON et A. ZSIGRI : A new way of doing multicast on wavelength-routed optical networks. Dans *ONDM (IFIP Optical Network Design and Modelling)*, 2004.
- [46] A. BOUDANI, A. GUITTON et B. COUSIN : GXcast : une généralisation du protocole Xcast. *ISDM (Informations, Savoirs, Décisions et Médiations)*, (13), 2004. Numéro spécial sur Majecstic.
- [47] A. BOUDANI, A. GUITTON et B. COUSIN : GXcast : une généralisation du protocole Xcast. Dans *Majecstic (Manifestation des jeunes chercheurs STIC)*, 2003. Prix du meilleur papier.
- [48] P. GARCIA, A. ZSIGRI et A. GUITTON : A multicast reinforcement learning algorithm for WDM optical networks. Dans *ConTEL (International Conference on Telecommunications)*, 2003.
- [49] A. GUITTON, R. MARIE et M. MOLNÁR : Adaptation d’algorithmes *multicast* aux réseaux optiques. Dans *AlgoTel (Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications)*, 2003.

- [50] A. GUITTON et R. MARIE : Nouvelles heuristiques de routage *multicast* dans les réseaux optiques basés sur WDM. Dans *Setit (Sciences Électroniques, Technologies de l'Information et des Télécommunications)*, 2003.
- [51] M. MOLNÁR, A. GUITTON et P. LEGUESDRON : Arbres de diffusion de délai minimal à partir d'informations incertaines. Dans *Setit (Sciences Électroniques, Technologies de l'Information et des Télécommunications)*, 2003.
- [52] A. ZSIGRI, A. GUITTON et M. MOLNÁR : Two multicast algorithms for sparse splitting capable networks. Dans *ONDM (IFIP Optical Network Design and Modelling)*, 2003.
- [53] A. ZSIGRI, A. GUITTON et M. MOLNÁR : Construction of light-trees for WDM multicasting under splitting capability constraints. Dans *ICT (International Conference on Telecommunications)*, 2003.

CHAPITRE

8

ACTIVITÉS D'ENCADREMENT

Dans ce chapitre, je détaille les travaux de chacun des doctorants que j'ai co-encadrés, ainsi que les travaux des étudiants que j'ai encadrés pendant leur stage de master recherche. Je conclus ce chapitre avec une réflexion sur ma méthode d'encadrement.

8.1 Co-encadrement de doctorants

Voici la liste des doctorants que j'ai co-encadrés.

01/10/2007-12/07/2011 : Co-encadrement de Mme Nassima HADID avec le Pr. Michel MISSON, sur le sujet : « Utilisation d'un canal d'aparté pour des applications de réseaux de capteurs embarquées », grâce à un financement MENRT. Qualifiée en section 27 du CNU en 2012.

01/02/2008-12/12/2011 : Co-encadrement de Mme Nancy EL RACHKIDY avec le Pr. Michel MISSON, sur le sujet : « *Cross-layering* et routage dans un réseau *ad-hoc* : politique de relais de trame sur un réseau de capteurs sans fil organisé selon une topologie en arbre », grâce à un financement du CNRS libanais. Qualifiée en sections 27 et 61 du CNU en 2012. Maître de conférences à Polytech Clermont-Ferrand, Université Blaise Pascal depuis septembre 2012.

01/10/2012-* : Co-encadrement de Mme Affoua Thérèse ABY avec le Pr. Michel MISSON, sur le sujet : « Réseaux de capteurs sans fil étendus dédiés aux collectes de données environnementales », grâce à un financement MENRT.

8.1.1 Co-encadrement de Nassima HADID

J'ai co-encadré la thèse de Nassima HADID avec le Pr. Michel MISSON du 1er octobre 2007 au 12 juillet 2011. Nassima a été financée par une bourse MENRT. Elle a travaillé sur le sujet : « Utilisation d'un canal d'aparté pour des applications de réseaux de capteurs embarquées ». Plus précisément, elle a considéré deux problématiques : l'impact de la période d'inactivité de IEEE 802.15.4 sur les communications, et la mise en place d'un protocole tolérant au délai fonctionnant sur un canal commun à plusieurs mobiles.

La période d'inactivité de IEEE 802.15.4 a un impact important sur les communications [24]. En effet, le trafic généré par l'application pendant la période d'inactivité est accumulé dans les files d'attente des nœuds. Lorsque la période d'activité des nœuds commence, ce trafic accumulé conduit à une forte sollicitation du médium. Nassima a donc proposé des protocoles permettant de s'adapter à cette forte sollicitation en début de période d'activité [19, 25]. Ces travaux concernent la partie 3.1 de ce document.

Ensuite, Nassima a proposé un protocole tolérant au délai pour des mobiles constitués de plusieurs nœuds, chaque mobile ayant son canal propre, et les mobiles communiquant entre eux sur un canal commun. Elle a proposé des optimisations des phases de détection des mobiles voisins et d'association [15]. Puis, elle a proposé des protocoles de communication intra-mobile et inter-mobiles, qui permettent d'utiliser, selon les besoins, le canal propre à chaque mobile ou le canal commun à tous les mobiles [9, 12]. Ces travaux sont mentionnés dans la partie 5.2.2.

Les travaux de thèse de Nassima ont fait l'objet des publications suivantes : [9, 12, 15, 19, 21, 24, 25].

Suite à sa thèse, Nassima a été qualifiée dans la section 27 du CNU. Elle travaille actuellement dans l'industrie.

8.1.2 Co-encadrement de Nancy EL RACHKIDY

J'ai co-encadré la thèse de Nancy EL RACHKIDY avec le Pr. Michel MISSON du 1er février 2008 au 12 décembre 2011. Nancy a été financée par une bourse du CNRS libanais. Elle a travaillé sur le sujet : « *Cross-layering* et routage dans un réseau *ad-hoc* : politique de relais de trame sur un réseau de capteurs sans fil organisé selon une topologie en arbre ». Plus précisément, les travaux de Nancy ont été menés selon deux axes : la réduction de la congestion en optimisant les protocoles MAC et routage, et la gestion des échanges de paquets entre files d'attente.

Nancy a proposé plusieurs protocoles pour réduire la congestion dans un réseau de capteurs sans fil. Certains de ces protocoles optimisent la phase de déploiement du réseau, en faisant en sorte que l'instant d'activation des nœuds dépende de la topologie du réseau [6, 18]. D'autres protocoles optimisent la phase de communications, en faisant en sorte que les messages suivent des routes éloignées pour aller à un même puits [2, 7, 20]. Des réductions significatives de la congestion peuvent être obtenues quand plusieurs puits existent, et que la sélection des puits est faite conjointement avec la construction des routes [8, 16]. Ces travaux sont mentionnés dans la partie 5.2.3.

Les échanges de paquets entre files d'attente ont permis d'améliorer le protocole Macari en réutilisant une partie des intervalles de temps perdus [10]. Toutefois, ces échanges de files d'attente peuvent causer l'apparition de boucles. Nancy a donc proposé des protocoles pour supprimer ces boucles [4, 13]. Cette problématique est centrale dans ce document, et se retrouve dans les parties 3.2, 3.3 et 4.2.

Les travaux de thèse de Nancy et leurs suites ont fait l'objet des publications suivantes : [2, 3, 4, 5, 6, 7, 8, 10, 13, 14, 16, 18, 20].

Suite à sa thèse, Nancy a été qualifiée dans les sections 27 et 61 du CNU. Nancy est actuellement maître de conférences à Polytech Clermont-Ferrand, Université Blaise Pascal, depuis septembre 2012.

8.1.3 Co-encadrement de Affoua Thérèse ABY

Je co-encadre la thèse d'Affoua Thérèse ABY avec le Pr. Michel MISSON depuis le 1er octobre 2012. Thérèse est financée par une bourse MENRT. Elle travaille sur le sujet : « Réseaux de capteurs sans fil étendus dédiés aux collectes de données environnementales ». Actuellement, elle se concentre sur la réalisation d'un protocole MAC économe en énergie et passant à l'échelle.

Thérèse travaille sur la conception et la spécification d'un protocole MAC distribué qui passe à l'échelle [1]. Dans l'état actuel des réflexions autour de ce protocole, l'activité de chaque nœud est très faible et est déterminée de manière aléatoire, ce qui a de nombreux avantages : aucune synchronisation n'est nécessaire, les rendez-vous entre nœuds se produisent avec une grande probabilité (au détriment du délai), et il est peu probable que de nombreux nœuds soient actifs simultanément (ce qui réduit la contention du médium). Ce protocole pourrait être utilisé dans la phase d'attente d'un évènement significatif. Ces travaux sont présentés dans les parties 4.1 et 4.2.

Les travaux que Thérèse doit entreprendre durant la suite de sa thèse consistent à approfondir les spécifications du protocole, à évaluer les performances de ce protocole au moyen d'un simulateur réseau, et à concevoir et tester un protocole de routage permettant aux nœuds de communiquer dans une grande topologie (avec des nœuds déployés de manière aléatoire dans un premier temps), rendue dynamique par les inactivités fréquentes des nœuds.

Les travaux de thèse de Thérèse ont fait l'objet de la publication suivante : [1].

8.2 Encadrement de stages de master recherche

Voici la liste des étudiants de master recherche que j'ai encadrés pendant leur stage.

2008 : Encadrement de M. Hamid ATTALAH, sur la synchronisation distribuée dans les réseaux de capteurs.

2009 : Encadrement de M. Hani MAHMOUDI, sur la synchronisation distribuée dans les réseaux de capteurs.

2011 : Encadrement de M. Michael ABDALLAH, sur l'association dans les réseaux de capteurs avec des conditions de propagation réalistes.

2013 : Encadrement de M. Rezzag ABDERRAHMEN, sur les réseaux de capteurs sans fil partiellement enfouis.

8.2.1 Encadrement de Hamid ATTALAH

J'ai encadré le stage de master recherche de Hamid ATTALAH, en 2008, sur l'étude des performances de la synchronisation dans les réseaux de capteurs. Ce sujet est issu des réflexions autour du projet Ocari (et plus particulièrement de l'optimisation de la période de synchronisation), et vise à quantifier et corriger la dérive des nœuds.

Hamid a étudié plusieurs protocoles distribués de synchronisation, en se concentrant sur la manière dont la dérive des nœuds pouvait être prise en compte de manière analytique. L'idée de son algorithme est de profiter de la propagation du message de synchronisation pour calculer le décalage entre les horloges des nœuds. En utilisant deux décalages successifs, son algorithme peut estimer la dérive entre les horloges des nœuds, et ainsi tenter de corriger cette dérive (ce qui augmente la précision du mécanisme de synchronisation, et réduit sa consommation énergétique). Hamid a évalué ses propositions sur une topologie en arbre. Ces travaux touchent des thématiques similaires à celles décrites dans 3.1.2.

8.2.2 Encadrement de Hani MAHMOUDI

J'ai encadré le stage de master recherche de Hani MAHMOUDI, en 2009, sur la synchronisation multi-sauts dans les réseaux de capteurs. Ce sujet est issu des réflexions autour du projet Ocari (et plus particulièrement de l'optimisation de la période de synchronisation), et vise à réduire certaines imprécisions temporelles.

Hani a étudié plusieurs protocoles distribués de synchronisation, en se concentrant sur la manière dont la synchronisation multi-sauts pouvait être réalisée sans collisions. Hani a conçu un protocole de synchronisation dans lequel les récepteurs se synchronisent sur les instants de réception d'un message (ce qui permet d'ignorer les imprécisions temporelles venant de l'émetteur). Son protocole peut être utilisé sur une topologie en arbre pour réaliser la synchronisation, tout en garantissant une bonne précision et en envoyant peu de messages de contrôle. Ces travaux touchent des thématiques similaires à celles décrites dans 3.1.2.

8.2.3 Encadrement de Michael ABDALLAH

J'ai encadré le stage de master recherche de Michael ABDALLAH, en 2011, sur l'association dans les réseaux de capteurs avec des conditions de propagation réalistes. Ce sujet est issu des réflexions autour du projet Ocari, et concerne le mécanisme d'association.

Michael a implémenté dans le simulateur réseau NS2¹ des conditions de propagation réalistes, en prenant en compte d'une part, des variations maîtrisées des caractéristiques d'un lien donné, et d'autre part, une grande variation des caractéristiques entre les liens (même lorsqu'ils sont de même longueur). Il a montré qu'avec

1. Et plus particulièrement, dans la version du simulateur sur laquelle notre équipe travaille, qui est décrite dans la partie 5.2.1.

cette modélisation réaliste des conditions de propagation, il était possible d'améliorer la procédure d'association des nœuds (en changeant les critères de sélection du nœud auquel s'associer). Ces travaux touchent des thématiques similaires à celles décrites dans 3.1.2.

8.2.4 Encadrement de Rezzag ABDERRAHMEN

J'ai encadré le stage de master recherche de Rezzag ABDERRAHMEN, en 2013, qui a travaillé sur les réseaux de capteurs sans fil partiellement enfouis. Ce sujet est issu des réflexions autour du projet Clervolc, et concerne l'évaluation des conditions de propagation quand les nœuds du réseau de capteurs se retrouvent recouverts (de végétation ou de terre).

Rezzag a développé un protocole de mesure de la qualité des liens radios sur la plateforme matérielle TelosB², et a réalisé des mesures dans trois environnements : dans le couloir d'un bâtiment (les couloirs agissant comme guides d'ondes, ils génèrent des conditions de propagation souvent très favorables), dans un environnement extérieur ouvert (ce qui correspond généralement à un cas de base), et dans une haie. Les mesures dans la haie, qui constituent la contribution principale de Rezzag, montrent une dégradation très nette des conditions de propagation, due à la traversée difficile de la végétation par les signaux à 2,4 GHz. Ces travaux nous conduisent à envisager pour les déploiements extérieurs des solutions réseaux basées sur des bandes de fréquences beaucoup plus basses, et notamment sur la bande de fréquences 433 MHz. Ces travaux sont décrits dans la partie 4.1.2.

8.3 Réflexions sur ma méthode d'encadrement

L'encadrement de doctorants représente une partie importante (à la fois au niveau de la mission et du temps passé) du métier d'enseignant-chercheur. Il est difficile de dresser une méthode précise d'encadrement, étant donné que chaque doctorant a sa propre personnalité, que chaque sujet de thèse a ses particularités, et que les travaux entrepris font continuellement évoluer le sujet de thèse. Toutefois, dans la suite, j'essaie de présenter la méthode générale que j'essaie d'adapter à chaque situation.

En début de thèse, je commence par présenter au doctorant le contexte dans lequel il se trouve, en présentant notamment le laboratoire LIMOS, l'équipe de recherche « réseaux et protocoles », le statut d'enseignant-chercheur et quelques instances nationales (par exemple l'AERES, l'ANR, le CNRS et le CNU). Ensuite, je lui explique le fonctionnement des journaux et des conférences, avec les procédures de soumission, de relecture et de notification, ainsi que les règles de publication. Puis, je lui explique la démarche envisagée pour la suite, en utilisant son sujet de thèse comme point central de la discussion.

Pour faciliter la prise en main des outils de développement que nous utilisons dans l'équipe (et notamment le simulateur NS2), j'ai rédigé des tutoriels qui sont à disposition sur un espace collaboratif interne à l'équipe. Je dispose aussi de tutoriels pour

2. Memsic : TelosB, http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf, septembre 2013.

quelques outils mathématiques que le doctorant est susceptible d'utiliser (comme le calcul d'intervalles de confiance, le calcul d'une fonction polynomiale permettant d'approcher une courbe, ou la programmation linéaire en nombres entiers).

Pendant la thèse, j'essaie de maintenir un suivi proche et technique du travail réalisé, en faisant des réunions techniques fréquentes (quitte à ce qu'elles soient courtes). Je participe activement au développement logiciel, en conseillant le doctorant sur l'architecture logicielle à employer, et en relisant les parties significatives du code source écrit (ce qui inclut souvent une participation active au débogage).

Je suis très présent lors de la rédaction du premier papier, afin de pouvoir expliquer en détail (et sur un exemple concret) la manière dont un article scientifique doit être écrit. Pour l'écriture des papiers suivants, je m'implique beaucoup sur le plan, qui me paraît être un point crucial dans tout document scientifique. J'essaie d'inciter le doctorant à être dans une logique de soumission régulière, ce qui a l'avantage de maintenir un certain niveau d'activité, et réduit l'abattement en cas de refus.

En fin de thèse, j'aide les doctorants à préparer leurs soutenances (notamment en faisant plusieurs répétitions). Après la soutenance, je les conseille sur leur recherche d'emploi, sur les procédures de qualification et de candidature, et sur la rédaction de dossiers. Je m'attache à ce que chacun de mes doctorants ait des chances raisonnables d'être qualifié en section 27 du CNU.

En synthétisant cette méthode d'encadrement, je me rends compte que certaines des pratiques que j'essaie de mettre en place sont proches des méthodes de développement dites agiles³. En perspective à moyen terme, j'envisage d'essayer d'approfondir l'utilisation de ces méthodes et des outils associés.

3. K. BECK, M. BEEDLE, A. VAN BENNEKUM, A. COCKBURN, W. CUNNINGHAM, M. FOWLER, J. GRENNING, J. HIGHSMITH, A. HUNT, R. JEFFRIES, J. KERN, B. MARICK, R. C. MARTIN, S. MELLOR, K. SCHWABER, J. SUTHERLAND, D. THOMAS : Manifesto for Agile Software Development, Agile Alliance, 2010.

CHAPITRE

9

GESTION DE LA RECHERCHE

Dans ce chapitre, je commence par détailler ma participation à des projets de recherche. Une partie importante de mes activités de recherche (et environ 70% de mes publications) est faite dans le cadre de ces projets. Ensuite, je donne des éléments concernant les communautés scientifiques auxquelles j'appartiens. Enfin, je décris mes autres activités ayant un lien avec la recherche.

9.1 Participation à des projets

Voici la liste des projets sur lesquels j'ai travaillé après ma thèse.

2005-2007 : Projet EPSRC TIME-EACM. Investissement : 100% du temps.

2007-2011 : Projet ANR Ocari. Investissement : 75% du temps recherche.

2012-* : Labex Clervolc. Participation à un programme transversal (sur la réalisation d'un réseau de capteurs). Investissement : 75% du temps recherche.

Dans la suite, je présente brièvement ces trois projets, qui correspondent à des applications différentes de surveillance.

9.1.1 Surveillance de routes (2005-2007)

Pendant mes post-doctorats, j'ai travaillé dans le cadre du projet TIME-EACM (*Transport Information Monitoring Environment - Event Architecture and Context Management*), financé par l'EPSRC (*Engineering and Physical Sciences Research Council*, qui est un organisme anglais de financement de projets de recherche scientifique). Ce projet consistait à surveiller le trafic routier d'une ville (en prenant la ville de Cambridge comme référence).

Notre équipe s'est concentrée sur la réalisation d'un réseau de capteurs sans fil et d'algorithmes adaptés à la surveillance du trafic routier. Des capteurs mesurent le nombre (et parfois la vitesse) de véhicules traversant une section de route (au moyen de câbles traversant la voie, de boucles à induction sous la chaussée, ou de doubles lasers placés sur le bas-côté). Les nœuds capteurs collectent les données sous forme de séries temporelles, les stockent et les transmettent à la demande à un puits applicatif, en multi-sauts. Le puits est capable d'interroger les nœuds capteurs pour leur demander leurs séries temporelles. Dans ce contexte, je me suis concentré sur deux thématiques : l'optimisation de multiples requêtes dynamiques venant du puits, et la compression des séries temporelles venant des capteurs.

L'optimisation de requêtes multiples consiste à mutualiser les requêtes ou les réponses, afin d'optimiser le routage des données ou leur agrégation. Dans [31], nous mettons en avant les relations qui existent entre routage et agrégation. Nous proposons plusieurs algorithmes distribués de construction d'arbres de collecte pour les données issues des capteurs, de manière à ce que les arbres aient de nombreuses parties communes. Dans [38], nous étudions plusieurs techniques d'agrégation lorsque les nœuds capteurs sont autorisés à transmettre des données approchées, l'enjeu étant de maximiser l'agrégation tout en conservant l'erreur introduite en dessous d'un certain seuil.

La compression de séries temporelles de trafic sur les nœuds capteurs permet de réduire le volume des messages échangés, ainsi que la taille mémoire occupée par les données. Dans [32], nous avons montré qu'il était possible de compresser significativement les données en utilisant une approche à base de transformée de Fourier ou d'ondelettes, et nous avons proposé des implémentations efficaces sur des nœuds capteurs. Nous avons aussi remarqué que le trafic routier génère de fortes corrélations temporelles et spatiales entre les nœuds capteurs proches (par rapport à la topologie routière), ce qui nous a permis d'augmenter le taux de compression en combinant une compression basée sur la transformée de Fourier et une compression basée sur les corrélations. Dans [29], nous étudions les avantages de la compression dans un environnement où les pertes sont fréquentes. Après avoir implémenté plusieurs protocoles de compression sur les nœuds capteurs, nous avons proposé un algorithme basé sur l'algorithme de Huffman adaptatif : notre algorithme utilise des acquittements par blocs et une mise à jour du dictionnaire périodique (plutôt que systématique). L'efficacité de notre algorithme lorsque les retransmissions sont interdites est semblable à celle des algorithmes basés sur les retransmissions sans compression.

Référence des publications autour de la thématique :

- Présentation du projet : [30].
- Optimisation de requêtes : [31, 33, 36, 38].
- Compression de données : [29, 32, 34].

Participation à la vie du projet :

- participation régulière aux réunions du projet,
- implémentation des programmes et protocoles proposés dans le projet, sur la plateforme TelosB, en NesC (100%),
- réalisation d'un utilitaire en Perl permettant de générer le squelette d'un programme NesC afin de simplifier le développement (100%),

- interfaçage d'un double laser (permettant de détecter la présence et la vitesse de voitures) avec une plateforme TelosB, en NesC (100%).

9.1.2 Surveillance de sites industriels (2007-2011)

J'ai été intégré dans le projet Ocari (Optimisations des Communications Ad-hoc pour les Réseaux Industriels), financé par l'ANR, suite à mon recrutement. Ce projet consistait à développer une pile protocolaire pour des réseaux de capteurs sans fil industriels à très faible mobilité (voir la partie 3.1).

Notre équipe s'est concentrée sur la réalisation d'un protocole d'accès au médium prenant en charge deux types de services : un trafic faible mais contraint en temps, et un trafic important mais non contraint en temps. Ce protocole d'accès au médium, nommé Macari (MAC pour Ocari) [27, 28], et présenté dans la partie 3.1.2, découpe le temps en périodes d'activités fonctionnant différemment. En complément de ma participation à la réalisation du protocole Macari, j'ai contribué à l'amélioration de l'utilisation des périodes d'activités.

L'optimisation de la période de synchronisation consiste à faire en sorte que des entités anticipent la transmission de leur balise de synchronisation, afin de réduire la durée de la synchronisation [26]. Cette anticipation ne peut se faire que si un nœud a le temps de préparer sa balise avant de la transmettre, et il est impératif que chaque nœud sache quels sont les nœuds qui anticipent leurs transmissions (afin que la synchronisation soit conservée pour tous ces nœuds). Une optimisation supplémentaire consiste à changer l'ordre dans lequel les nœuds sont listés, afin de maximiser le gain de l'anticipation [11]. D'autres manières de synchroniser les nœuds ont été étudiées par Hamid ATTALAH et Hani MAHMOUDI pendant leur stage de master recherche.

L'optimisation de la période d'activités ordonnancées consiste à paralléliser les activités des nœuds en utilisant une technique de réutilisation spatiale, implémentée grâce à un algorithme de coloriage à trois sauts [17] pouvant supporter les changements de topologie et les liens unidirectionnels. Pour gérer les conflits de couleurs (qui peuvent survenir suite à des changements de conditions de propagation, à l'ajout de nœuds, ou pendant les phases d'établissement du réseau), nous avons proposé un protocole d'accès au médium supportant des conflits de couleur, mais n'introduisant aucun délai supplémentaire en l'absence de conflits. Une autre optimisation consiste à permettre aux nœuds d'utiliser les paquets prévus pour la période d'activités non ordonnancées lorsque les files d'attente pour la période d'activités ordonnancées sont vides : c'est sur cet aspect qu'a travaillé Nancy EL RACHKIDY pendant sa thèse (voir les parties 3.2 et 3.3).

L'optimisation de la période d'activités non ordonnancées consiste à améliorer les performances du protocole CSMA/CA slotté. L'amélioration est faite en agissant sur de nombreux leviers : le choix précis des délais introduits lors des tentatives d'accès au médium [19], la simplification de la procédure d'association [15], ou encore la réduction des délais dus aux reports de trames [24]. Ces travaux regroupent une partie importante des activités de Nassima HADID au cours de sa thèse.

Référence des publications autour de la thématique :

- Présentation du projet : [22, 23].

- Accès au médium : [11, 17, 19, 21, 24, 25, 26, 27, 28].
- Échange de files d’attente : [10, 13].

Participation à la vie du projet :

- participation régulière aux réunions du projet,
- co-rédaction de livrables (notamment, du SP5),
- participation à la démonstration finale du SP5,
- implémentation, en Perl, d’un logiciel permettant de tracer graphiquement le comportement temporel de la sous-couche MAC à partir du journal d’exécution du prototype, obtenu par capture (100%),
- réalisation d’une interface homme machine en Java permettant de communiquer avec les nœuds capteurs par le port série (100%),
- adaptation de NS2 pour une meilleure prise en compte des réseaux de capteurs sans fil : co-implémentation de la prise en compte réaliste de la partie matérielle (dérive des horloges, temps de transfert entre le module radio et le processeur) et des couches basses (propagation, couche physique et sous-couche MAC) dans le simulateur NS2 (50%),
- écriture de générateurs de topologie en Perl (100%),
- dépôt de code APP (dépôt conjoint entre le LIMOS et l’IRIT), et mise sous licence GPLv3 du code de prototypage (5%).

9.1.3 Surveillance de sites environnementaux (2012-*)

Je participe au Labex Clervolc (Centre Clermontois de Recherche sur le Volcanisme) depuis 2012. Ce Labex consiste à améliorer la connaissance des volcans et à mieux gérer les risques volcaniques (voir la partie 4.1).

Notre équipe se concentre sur la conception d’une architecture réseau permettant à un réseau de capteurs sans fil de surveiller un volcan ou une grande étendue. Le réseau doit donc être adapté à un déploiement en extérieur agressif (intempéries, fumerolles, éboulements, etc.). La surveillance du volcan est faite par phases : quand le volcan est au repos, des mesures régulières et peu fréquentes sont réalisées, et les nœuds capteurs économisent de l’énergie en adoptant un rythme d’activités très faible ; quand le volcan se réveille, le réseau passe dans un mode de surveillance intensive et des mesures sont effectuées et transmises très fréquemment.

L’adaptation du réseau aux conditions environnementales est l’un des défis du projet. Cette adaptation doit se faire à de nombreux niveaux. Par exemple, le matériel des nœuds capteurs doit être tropicalisé. La couche physique doit être adaptée aux conditions environnementales (nœuds capteurs partiellement enfouis, présence de végétation, absence de ligne de vue entre nœuds capteurs) : c’est sur ce point qu’a travaillé Rezzag ABDERRAHMEN pendant son stage de master (voir la partie 4.1.2). La sous-couche MAC et la couche réseau (et plus généralement, l’architecture logique du réseau) doivent être conçues pour être robustes et redondantes : c’est le point essentiel de la thèse d’Affoua Thérèse ABY (voir les parties 4.1.3 et 4.1.4).

La surveillance du volcan par phases conduit à imaginer des protocoles spécifiques par phase (par exemple, certains protocoles sont adaptés à des communications occasionnelles, alors que d’autres sont adaptés à des communications fréquentes). Lorsque des changements de phases interviennent (à cause de la détection d’évènements exté-

rieurs ou du traitement d'une commande envoyée par un opérateur), des précautions doivent être prises pour que le changement de protocoles ne cause pas une baisse significative des performances (même si cette baisse n'est que temporaire).

Référence des publications autour de la thématique :

- Évitement des boucles pendant les changements de phases : [4, 5].
- Conception d'un protocole MAC permettant le passage à l'échelle : [1].

Participation à la vie du projet :

- participation régulière aux réunions du projet,
- co-rédaction (avec le Dr Philippe LABAZUY du laboratoire LMV) du lot 1 (réseaux de capteurs intelligents) du projet AMI Géothermia ; partie scientifique acceptée et projet en attente de financement (budget d'environ 950 k€ pour le lot 1),
- recrutement d'un ingénieur pour développer une station de base générique et multi-interfaces à déployer au sommet d'un volcan (50%).

9.2 Communautés scientifiques

Dans cette partie, je présente mes collaborations internationales, puis mon implication dans des conférences internationales (au niveau de l'organisation, du comité scientifique ou des relectures), et finalement ma participation à des jurys de thèse.

9.2.1 Collaborations

Au niveau international et depuis mon recrutement à Clermont-Ferrand, je collabore principalement avec les personnes suivantes.

- Dr Chiara BURATTI, maître de conférences à l'Université de Bologne (en Italie). Nous travaillons sur les protocoles de routage [2], et sur les liens entre simulation et prototypage [3].
- Dr Waltenegus DARGIE, maître de conférences habilité à l'Université Technique de Dresden (en Allemagne). Nous avons travaillé sur l'optimisation de la synchronisation [11], et sur les protocoles de routage pour les réseaux linéaires.
- Pr. Gilles DELISLE, professeur à l'Université de Laval (au Canada), et Dr Nadir HAKEM, maître de conférences à l'Université de Québec en Abitibi-Témiscamingue (au Canada). Nous travaillons sur la conception de réseaux de capteurs sans fil adaptés aux déploiements sur sites naturels (à la fois au niveau du matériel et des couches basses de la pile protocolaire).
- Pr. Abraham FAPOJUWO, professeur à l'Université de Calgary (au Canada). Nous avons travaillé sur l'algorithme de contention CSMA/CA pour les réseaux de capteurs sans fil.

Au niveau national et depuis mon recrutement à Clermont-Ferrand, j'ai collaboré principalement avec les personnes suivantes.

- Dr Pascale MINET, chargée de recherche habilitée à diriger des recherches à Inria Rocquencourt. Nous avons travaillé sur le *cross-layering* et la réutilisation temporelle d'intervalles de temps.
- Pr. Thierry VAL, professeur à l'Université de Toulouse, et Dr Adrien VAN DEN BOSSCHE, maître de conférences à l'Université de Toulouse. Nous avons

travaillé sur les protocoles MAC, le *cross-layering* et l’affinage de la simulation par retour d’expérience de prototypage.

Au niveau local et depuis mon recrutement à Clermont-Ferrand, je collabore principalement avec les personnes suivantes (hormis les membres de l’équipe).

- Dr Karim KELFOUN, maître de conférences au LMV. Nous travaillons sur la réalisation d’un réseau de capteurs sans fil pour la surveillance d’un volcan.
- Dr Philippe LABAZUY, maître de conférences à l’OPGC (Observatoire de Physique du Globe à Clermont-Ferrand). Nous travaillons sur la réalisation d’un réseau de capteurs sans fil pour la surveillance d’une centrale géothermique.
- Dr Olivier RAYNAUD, maître de conférences au LIMOS. Nous travaillons sur les propriétés du treillis modélisant l’évolution d’un réseau.

9.2.2 Organisation de manifestations scientifiques

2012 : ICWCUCA (*International Conference on Wireless Communications in Unusual and Confined Areas*), à Clermont-Ferrand, France : membre du comité de pilotage, co-responsable du TPC, interlocuteur auprès de l’IEEE.

2011 : HeterWMN (*6th IEEE International Workshop on Heterogeneous, Multi-Hop, Wireless and Mobile Networks*) en conjonction avec IEEE Globecom, à Houston, États-Unis : co-organisateur.

2009 : L2I (journées de travail sur les réseaux sans fil des laboratoires LIMOS, LIRMM et IRIT), à Clermont-Ferrand, France : co-organisateur de l’instance locale.

9.2.3 Comités scientifiques de conférences (hors organisation)

Adoc : association des doctorants de l’école doctorale MATISSE, responsable du comité de relecture, 2004 et 2005.

Algotel : rencontres francophones sur les aspects algorithmiques des télécommunications, 2014.

AINA : *IEEE International Conference on Advanced Information Networking and Applications*, 2013.

Casemans : *ACM International Workshop on Context Awareness for Self-Managing Systems (Devices, Applications and Networks)*, en conjonction avec ACM Ubicomp, 2010 et 2011.

HeterWMN : *IEEE International Workshop on Heterogeneous, Multi-Hop, Wireless and Mobile Networks*, en conjonction avec IEEE Globecom, 2010 et 2012.

ICCCN : *International Conference on Computer Communications and Networks*, 2014.

ICIT : *IEEE International Conference on Industrial Technology*, 2014.

IJERTCS : *International Journal of Embedded and Real-Time Communication Systems*, numéro spécial nommé *self-aware and adaptive embedded systems*, 2011.

- ISCAIE** : *IEEE Symposium on Computer Applications and Industrial Electronics*, 2014.
- ISWTA** : *IEEE Symposium on Wireless Telecommunications Applications*, 2012 et 2013.
- MIC** : *Mosharaka International Conference*, plusieurs conférences : MIC-WCMC 2012 (*Wireless Communications and Mobile Computing*), session WCN (*Wireless Communication Networks*); MIC-CES 2011 (*Communications and Electronic Systems*), session MAS (*Mesh, Ad Hoc and Sensor Networks*); MIC-CSP 2011 (*Communications and Signal Processing*), session CNA (*Communication Networks and Applications*); MIC-CNIT 2010 (*Communications, Networking and Information Technology*), session CIT (*Communication and Information Technology*).
- PIMRC** : *IEEE Personal, Indoor and Mobile Radio Communications* : session MAC (*MAC and Cross-Layer Design*) en 2012 et 2013; session *Mobile and Wireless Networks* en 2013.
- PMECT** : *IEEE International Workshop on Performance Modeling and Evaluation of Computer and Telecommunication Networks*, en conjonction avec IEEE ICCCN, de 2010 à 2013.
- SANES** : *Self-Adaptive Networked Embedded Systems*, session spéciale de PECCS (*International Conference on Pervasive and Embedded Computing and Communication Systems*), 2012 et 2013.

9.2.4 Relectures (hors comités scientifiques)

- 2013** : Journaux DAEM (*Design Automation for Embedded Systems*), *Computer Communications*, IEEE TPDS (*Transactions on Parallel and Distributed Systems*), IEEE Sensors, JNCA (*Journal of Computer Networks and Applications*) et JCM (*Journal of Communications*). Conférences IEEE Globecom, IEEE ISCI (*IEEE Symposium on Computers & Informatics*), IEEE WCNC (*Wireless Communications and Networking Conference*).
- 2012** : Journaux IEEE TPDS, IEEE Sensors, IEEE *Communications Letters*, *Computer Communications*, *International Journal on Communication Networks and Distributed Systems*, AHSWN (*Ad Hoc & Sensor Wireless Networks*) et JSAN (*Journal of Sensor and Actuator Networks*). Conférences IEEE VTC (*Vehicular Technology Conference*) et ICT (*International Conference on Telecommunications*).
- 2011** : Journal IEEE *Sensors*. Conférence IEEE VTC.
- 2010** : Journaux IEEE *Sensors*, IEEE TVT (*Transactions on Vehicular Technology*), MONE (*Mobile Networks and Applications*), JNW (*Journal of Networks*) et MEDJCN (*Mediterranean Journal of Computers and Networks*). Conférence COSI.
- 2009** : Journaux JNW (*Journal of Networks*), PMC (*Pervasive and Mobile Computing*) et TKDE (*Transactions on Knowledge and Data Engineering*).
- 2008** : Journaux JNW et PMC. Conférences CARI et CSR.

2007 : Journal IEEE *Communications Letters*.

2006 : Conférence IEEE Globecom.

2005 : Conférence IFIP Networking.

2004 : Journal *Computer Communications*. Conférences IEEE ISCC et IFIP Networking.

2003 : Journal *Computer Networks*. Conférence RenPAR.

9.2.5 Jurys de thèse (hors co-direction)

19/12/2013 : Examineur de la thèse de Maria Eugenia BEREZIN, à l'Université de Grenoble.

6/12/2013 : Examineur de la thèse de Saïd HARCHI, à l'Université de Lorraine (Nancy).

26/06/2013 : Examineur de la thèse de Réjane DALCÉ, à l'Université du Mirail (Toulouse).

7/12/2009 : Examineur de la thèse de Gérard CHALHOUB, à l'Université Blaise Pascal (Clermont-Ferrand).

7/5/2009 : Examineur de la thèse de Sabri BENFERHAT, à l'Université Blaise Pascal (Clermont-Ferrand).

9.3 Autres activités ayant un lien avec la recherche

2013 : Membre du comité de recrutement d'un maître de conférences 27ème/61ème section à l'Université Technologique de Troyes (pôle ROSAS).

2010 : Membre du comité de recrutement d'un maître de conférences 27ème section à Polytech Clermont-Ferrand (laboratoire LIMOS).

2009 : Membre du comité de recrutement de deux maîtres de conférences 27ème section à l'IUT de Lannion (laboratoire IRISA).

2009 : Membre du comité de recrutement de deux maîtres de conférences 27ème section à l'UFR Sciences et Technologies (laboratoire LIMOS).

2008-* : Membre du comité d'experts 27ème section du LIMOS. Le comité d'experts propose la constitution du jury des comités de recrutement.

CHAPITRE

10

ACTIVITÉS PÉDAGOGIQUES

Mon parcours d'enseignement est le suivant.

2007-* : Maître de conférences à l'UFR Sciences et Technologies de l'Université Blaise Pascal (au moins 192h par an).

2005-2007 : Vacataire à *Birkbeck College, University of London* (6h par an).

2002-2005 : Moniteur de l'enseignement supérieur à l'Université de Rennes I (64h par an).

En parallèle de mon poste à l'Université Blaise Pascal, j'ai aussi effectué des vacances dans les établissements suivants :

- Isma (école privée) à Douala, Cameroun, pendant un an.
- Suptem (école privée) à Safi et Tanger, Maroc, pendant trois ans

10.1 Liste des enseignements

J'ai regroupé mes cours en trois pôles : réseau et système, programmation, et divers.

10.1.1 Cours de réseau et de système

L3 : réseaux : Protocoles Internet. 18h de CM, 6h de TD et 3h de TP par an pendant 3 ans.

L3 : systèmes d'exploitation : Fonctionnement d'un système d'exploitation. 18h de CM par an pendant 2 ans.

M2 : administration réseau : Métier d'administrateur réseau. Environ 25h CM, 5h TD et 5h TP par an pendant 6 ans.

M2 : architecture réseau : Conception d'un réseau. Environ 25h de CM par an pendant 7 ans.

M2 : network security : Sécurité (en anglais). 20h CM et 10h TP par an pendant 4 ans.

M2 : réseau avancé et virtualisation : Détails sur TCP et VLAN. 4h de CM et 10h de TP par an pendant 2 ans.

M2 : sécurité : Sécurité. 10h CM et 10h TD par an pendant 1 an.

10.1.2 Cours de programmation

PES : informatique : Utilisation d'un *shell* et programmation en Perl. 18h de CM et 15h de TP par an pendant 2 ans.

L1 : algorithmique : Algorithmique et programmation en C. Environ 30h CM, 12h TD et 3h TP par an pendant 5 ans.

L1 : programmation fonctionnelle : Programmation en Caml. Environ 40h de CM/TD par an pendant 3 ans.

L1 : programmation fonctionnelle : Programmation en Mathematica. 22h TD par an pendant 2 ans.

L2 : architecture des systèmes : Programmation en assembleur x86. 20h de CM par an pendant 2 ans.

L2 : programmation impérative : Programmation en Pascal. Environ 12h TD et 41h TP par an pendant 3 ans.

L2 : programmation orientée objets : Programmation en Java. Environ 14h de CM, 2h de TD et 7h de TP par an pendant 5 ans.

M2 : mobile computing : Programmation en NesC (en anglais). 6h de TP par an pendant 2 ans.

10.1.3 Autres cours

L1 : représentation des données : Représentation des données et encodage. 30h de CM par an pendant 4 ans.

L3 : bureautique : Traitement de textes et bases de données. Environ 21h de TP par an pendant 3 ans.

10.2 Activités administratives liées à l'enseignement

Dans la liste qui suit, les dates sont indiquées par années universitaires (plutôt que par années civiles).

2011-2013 : Membre du jury de recrutement de technicien de police technique et scientifique.

- 2010-2013** : Membre de la commission des études de l'UFR Sciences et Technologies.
- 2009-*** : Directeur d'études de première année pour l'informatique.
- 2008-2013** : Membre du conseil du département mathématiques et informatique de l'UFR Sciences et Technologies.
- 2008-2011** : Responsable pédagogique de la L2 informatique, incluant l'année 2008-2009 de transition du LMD1 au LMD2.
- 2008-2011** : Directeur d'études de première année pour la PES (préparation aux études scientifiques), incluant l'année de création 2008-2009.
- 2000-2002** : Secrétaire (1 an) puis président (1 an) de l'AUREFI (association universitaire rouennaise des étudiants de la filière informatique).

10.3 Activités annexes liées à l'enseignement

Dans la liste qui suit, les dates sont indiquées par années universitaires (plutôt que par années civiles).

- 2013-*** : Membre de la commission informatique de l'UFR.
- 2012-*** : Référent MIPE (opération micro portable étudiant) pour l'UFR.
- 2011-2012** : Formateur de la partie réseau de l'option ISN (informatique et sciences du numérique), pour les enseignants de terminale.
- 2011-2013** : Atelier sur la sécurité informatique pour des collégiens en immersion sur le campus, dans le cadre de MathC2+ (organisé par l'IREM (institut de recherche en enseignement des mathématiques)).
- 2010-2013** : Rédacteur du sujet d'informatique du concours de commissaire de police de la police nationale, et du sujet de technologies de l'information et de la communication du concours externe d'officier de police de la police nationale.
- 2010-2012** : Organisateur de la partie technique d'un concours de programmation d'un jeu vidéo, avec le bureau des étudiants de maths-info Enigma.
- 2008-*** : Atelier sur la conception de jeux vidéos pour la fête de la science.

Troisième partie

Annexes

ANNEXE

A

COMPLÉMENTS SUR LE GRAPHE D'ÉVOLUTION

Dans cette partie, nous décrivons quelques résultats supplémentaires sur le graphe d'évolution qui ne sont pas directement utiles pour la compréhension de ce document, mais qui permettent de mieux cerner certains points traités.

A.1 Compléments sur la structure du graphe d'évolution

Évolution multiple.

Définition 20 (Évolution multiple). L'évolution multiple correspond au cas où tous les voisins d'un nœud ayant changé d'état peuvent changer simultanément d'état.

L'évolution multiple permet de modéliser l'évolution du réseau lorsque le temps de propagation et les variations de temps de traitement des nœuds sont négligées, et lorsque tous les voisins d'un nœud sont actifs simultanément¹. Avec cette hypothèse, lorsqu'un nœud prévient qu'il a changé d'état, c'est par un message de diffusion : tous les voisins du nœud, s'ils sont réveillés, peuvent recevoir ce message et changer d'état simultanément.

Propriété 19. *Soit G un graphe connexe et a une graine. Quand l'évolution est multiple, mono-graine et bi-états, le graphe d'évolution $\mathcal{T}(G, a)$ forme un treillis.*

1. Le cas où seulement un sous-ensemble des voisins peut changer d'état simultanément est laissé en perspective.

Démonstration courte. La démonstration de cette preuve est similaire à celle du théorème 4. Elle s'obtient en partant de deux états distincts, X et Y , de $\mathcal{T}(G, a)$, et en montrant que $X \cup Y \in \mathcal{T}(G, a)$. La différence avec la preuve du théorème 4 est qu'à chaque saut d'un chemin, on ajoute un ensemble de nœuds (qui est l'ensemble des voisins d'un même nœud, plutôt que le nœud lui-même). Si on note y_i le nœud correspondant au i -ème saut du chemin de \emptyset à Y (tel que tous les voisins de y_i appartiennent à l'état atteint au $(i + 1)$ -ème saut du chemin), on peut montrer que $X \cup Y$ est atteint en suivant le chemin de \emptyset à X , puis en ajoutant à tour de rôle tous les voisins des nœuds y_i . La preuve se termine en montrant que $\mathcal{T}(G, a)$ contient l'ensemble vide et est fermé par union. \square

Évolution multi-états.

Définition 21 (Évolution multi-états). L'évolution multi-états correspond au cas où chaque nœud peut être dans différents états, numérotés séquentiellement à partir de 1. Tous les nœuds sont initialement dans l'état 1, et un nœud qui passe dans un état donné ne peut pas revenir dans un état inférieur². Le changement d'état suit les règles suivantes :

- la graine passe d'un état 1 à un état $k \geq 2$ arbitrairement,
- tout nœud peut passer d'un état $k \geq 2$ à un état $k' > k$ arbitrairement,
- quand un nœud n change d'état par l'intermédiaire de son voisin v , le nouvel état de n est égal à l'état de v .

Propriété 20. Soit G un graphe connexe et a une graine. Quand l'évolution est unitaire, mono-graine et multi-états, le graphe d'évolution $\mathcal{T}(G, a)$ forme un treillis.

Démonstration courte. La démonstration reprend celle du théorème 4, en changeant légèrement la représentation des états de $\mathcal{T}(G, a)$. Dans le cas multi-états, un état E est représenté comme un ensemble tel que : E contient $\{n_2, \dots, n_i\}$ si et seulement si le nœud n est dans l'état i . Par exemple, si les nœuds du graphe sont a, b, c et d , l'état $E = (b_2 b_3 d_2)$ correspond au nœud a dans l'état 1, b dans l'état 3, c dans l'état 1 et d dans l'état 2. \square

A.2 Propriétés du graphe d'évolution

Les propriétés données dans cette partie utilisent les hypothèses de base, à savoir que l'évolution est unitaire, mono-graine et bi-états dans un graphe connexe.

Propriété 21. Le treillis $\mathcal{T}(G, a)$ est borné.

Démonstration courte. L'élément minimum est \emptyset , et l'élément maximum est l'ensemble de tous les nœuds de G (car G est connexe). \square

2. Il est possible que l'automate représentant l'état des nœuds contienne des boucles, et donc qu'un nœud puisse revenir dans un état du passé. Dans la modélisation de l'évolution multi-états (comme dans l'évolution bi-états), nous considérons que les états ne font que progresser. Pour modéliser un automate qui aurait une boucle (e_1, \dots, e_n, e_1) , il suffit de considérer deux évolutions multi-états successives (la première de e_1 à e_n et la deuxième de e_n à e_1).

Propriété 22. *Le treillis $\mathcal{T}(G, a)$ est gradué, et la graduation d'un état correspond au nombre de nœuds qui sont dans l'état 2.*

Démonstration. Soit E un état. Notons $|E|$ le nombre de nœuds de E (ces nœuds sont dans l'état 2). Par construction du treillis selon l'évolution unitaire³, les successeurs d'un état E sont les états E' tels que $E' = E \cup \{v\}$, où v est le voisin d'un état $x \in E$, avec $v \notin E$. On a donc $|E'| = |E| + 1$, et $|\emptyset| = 0$. \square

On peut remarquer que l'ensemble des états de $\mathcal{T}(G, a)$ n'est pas fermé par intersection. Pour montrer un contre-exemple, considérons pour graphe G un cycle de 4 nœuds a, b, c et d , avec la graine a . Les états (abc) et (acd) existent dans $\mathcal{T}(G, a)$. En revanche, l'état $(abc) \cap (acd) = (ac)$ n'est pas un état de $\mathcal{T}(G, a)$.

On peut remarquer que le treillis $\mathcal{T}(G, a)$ n'est pas distributif. Pour montrer un contre-exemple, considérons à nouveau pour graphe G un cycle de 4 nœuds a, b, c et d , avec la graine a . Le treillis $\mathcal{T}(G, a)$ est représenté sur la figure A.1. Nous allons montrer que la propriété de distributivité n'est pas vérifiée pour un certain triplet d'états A, B et C . En effet, en posant $A = (abc)$, $B = (ab)$ et $C = (acd)$, nous avons d'une part $A \wedge (B \vee C) = (abc) \wedge ((ab) \vee (acd)) = (abc) \wedge (abcd) = (abc)$, et d'autre part $(A \wedge B) \vee (A \wedge C) = ((abc) \wedge (ab)) \vee ((abc) \wedge (acd)) = (ab) \vee (a) = (ab)$. Comme $A \wedge (B \vee C) \neq (A \wedge B) \vee (A \wedge C)$, le treillis $\mathcal{T}(G, a)$ n'est pas distributif⁴.

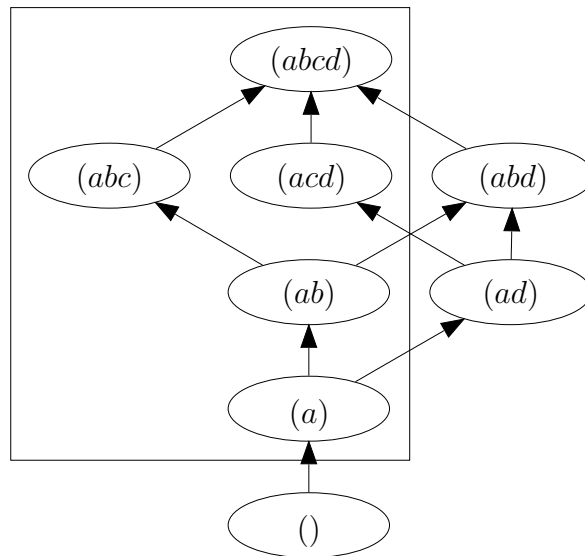


Figure A.1 – Le treillis $\mathcal{T}(G, a)$ n'est pas distributif : le treillis $\mathcal{N}5$ (encadré) est un sous-treillis de $\mathcal{T}(G, a)$.

Propriété 23. *Le treillis $\mathcal{T}(G, a)$ est semi-modulaire (résultat connu dans la littérature⁵).*

3. On peut remarquer que la propriété de graduation n'est pas vérifiée lorsque l'évolution est multiple ou multi-états.

4. Une autre preuve utilise le treillis nommé $\mathcal{N}5$, représenté dans le rectangle sur la figure A.1. Pour tout couple d'états A et B de $\mathcal{N}5$, l'état $A \vee B$ de $\mathcal{T}(G, a)$ appartient à $\mathcal{N}5$, et l'état $A \wedge B$ de $\mathcal{T}(G, a)$ appartient à $\mathcal{N}5$. Ainsi, $\mathcal{N}5$ est un sous-treillis de $\mathcal{T}(G, a)$, et $\mathcal{T}(G, a)$ n'est pas distributif.

5. A. KEZDY, S. SEIF : When is a poset isomorphic to the poset of connected induced subgraphs of a graph?. *Southwest Journal of Pure and Applied Mathematics*, 1:42–50, 1996.

Démonstration. Montrons que pour tout couple d'états A et B de $\mathcal{T}(G, a)$, si A est un successeur (direct) de $A \wedge B$, alors $A \vee B$ est un successeur (direct) de B . Soit A et B deux états de $\mathcal{T}(G, a)$ tels que A est un successeur de $A \wedge B$. Alors, il existe un unique nœud $x \notin B$ de G tel que $A = (A \wedge B) \cup \{x\}$. $A \vee B = A \cup B = ((A \wedge B) \cup \{x\}) \cup B = ((A \wedge B) \cup B) \cup \{x\} = B \cup \{x\}$, avec $x \notin B$. $A \vee B$ est bien un successeur direct de B . Le treillis $\mathcal{T}(G, a)$ est donc semi-modulaire. \square

En revanche, le treillis $\mathcal{T}(G, a)$ n'est pas modulaire. $\mathcal{T}(G, a)$ est modulaire si et seulement si pour tout triplet d'états E_1, E_2 et E_3 , si $E_3 \leq E_1$, $E_1 \wedge E_2 = E_3 \wedge E_2$ et $E_1 \vee E_2 = E_3 \vee E_2$, alors $E_1 = E_3$. Pour montrer un contre-exemple, considérons pour graphe G un cycle de 4 nœuds a, b, c et d , et la graine a . Le treillis $\mathcal{T}(G, a)$ est représenté sur la figure A.1. Si l'on pose $E_1 = (abc)$, $E_2 = (acd)$ et $E_3 = (ab)$, nous avons :

- $E_3 \leq E_1$ car $(ab) \subset (abc)$,
- $E_1 \wedge E_2 = E_3 \wedge E_2$ car $(abc) \cap (acd) = (a)$ et $(ab) \cap (acd) = (a)$,
- $E_1 \vee E_2 = E_3 \vee E_2$ car $(abc) \cup (acd) = (abcd)$ et $(ab) \cup (acd) = (abcd)$.

Comme $E_1 \neq E_3$, le treillis $\mathcal{T}(G, a)$ n'est pas modulaire.

Propriété 24. *Le treillis $\mathcal{T}(G, a)$ est complet (résultat connu dans la littérature⁶).*

Démonstration. Comme $\mathcal{T}(G, a)$ est fini et non vide, $\mathcal{T}(G, a)$ est complet. \square

On peut remarquer que le treillis $\mathcal{T}(G, a)$ n'est pas complémenté. $\mathcal{T}(G, a)$ est complémenté si et seulement si pour tout état X , il existe Y tel que $X \wedge Y = \perp$ et $X \vee Y = \top$. Pour montrer un contre-exemple, considérons pour graphe G un cycle de 4 nœuds a, b, c et d , avec la graine a . Le treillis $\mathcal{T}(G, a)$ est représenté sur la figure A.1. Si l'on pose $X = (abd)$, nous avons :

- aucun Y parmi $\{\emptyset, (a), (ab), (ad), (abd)\}$ ne satisfait la propriété car $X \vee Y = (abd) \cup Y = (abd)$, ce qui est différent de $(abcd)$,
- $Y = (abc)$ ne satisfait pas la propriété car $X \wedge Y = (abd) \wedge (abc) = (ab)$, ce qui est différent de \emptyset ,
- $Y = (acd)$ ne satisfait pas la propriété car $X \wedge Y = (abd) \wedge (acd) = (ad)$, ce qui est différent de \emptyset ,
- $Y = (abcd)$ ne satisfait pas la propriété car $X \wedge Y = (abd) \wedge (abcd) = (abd)$, ce qui est différent de \emptyset .

Ainsi, $X = (abd)$ n'a pas de complémentaire, et le treillis n'est pas complémenté.

Le treillis $\mathcal{T}(G, a)$ n'est pas atomistique, c'est-à-dire que certains des états sont des sup-irréductibles (dit autrement, ces états n'ont qu'un seul prédécesseur) mais ne sont pas des atomes (c'est-à-dire qu'ils ne sont pas successeurs de l'état \perp). Pour montrer un contre-exemple, considérons pour graphe G un cycle de 4 nœuds a, b, c et d , avec la graine a . Le treillis $\mathcal{T}(G, a)$ est représenté sur la figure A.1. On peut remarquer que l'état (abc) est un sup-irréductible ((ab) est son seul prédécesseur), mais n'est pas un atome, donc le treillis n'est pas atomistique.

De plus, le treillis $\mathcal{T}(G, a)$ n'est pas co-atomistique, c'est-à-dire que certains des états sont des inf-irréductibles (dit autrement, ces états n'ont qu'un seul successeur) mais ne sont pas des co-atomes (c'est-à-dire qu'ils ne sont pas prédécesseurs de

6. Voir la note de bas de page 5.

l'état \top). Pour montrer un contre-exemple, considérons le graphe G représenté sur la figure 4.12, avec la graine a . Le treillis $\mathcal{T}(G, a)$ est représenté sur la figure 4.14. On peut remarquer que l'état (ad) est un inf-irréductible (abd est son seul successeur), mais n'est pas un co-atome, donc le treillis n'est pas co-atomistique.

Propriété 25. *Soit G un graphe (connexe) de n nœuds et a une graine. Le nombre d'états de $\mathcal{T}(G, a)$ est compris entre $n + 1$ et $2^{n-1} + 1$. Ces bornes sont atteintes pour certains graphes G .*

Démonstration. Le nombre d'états de $\mathcal{T}(G, a)$ est supérieur ou égal à $n + 1$. En effet, dans le pire des cas, chaque état de $\mathcal{T}(G, a)$ ne contient qu'un seul successeur (sauf pour l'état \top). Cela se produit uniquement quand G est un chemin dont a est l'une des extrémités. Le nombre d'états de $\mathcal{T}(G, a)$ est égal au nombre de chemins de $\mathcal{T}(G, a)$ ayant a pour extrémité, plus l'état \emptyset , ce qui donne le résultat attendu.

Le nombre d'états de $\mathcal{T}(G, a)$ est inférieur ou égal à $2^{n-1} + 1$. En effet, si on note V l'ensemble des nœuds de G , les états de $\mathcal{T}(G, a)$ contiennent l'état \emptyset , ainsi que toutes les combinaisons (dans le pire des cas) de sous-ensembles de $V \setminus \{a\}$ (le nœud a étant ajouté à chacun de ces sous-ensembles). Il y a 2^{n-1} combinaisons de nœuds de $V \setminus \{a\}$, ce qui donne le résultat attendu. On peut remarquer que si G est une clique, ou une étoile centrée sur la graine a , on obtient le nombre d'états maximal. \square

Propriété 26. *Soit G un graphe de n nœuds et a une graine. Soit \mathcal{E}_m l'ensemble des états de $\mathcal{T}(G, a)$ qui contiennent m nœuds. Soit h le cardinal du plus grand ensemble de nœuds X tel que le graphe induit par X est un chemin ayant a comme extrémité. Alors, pour tout $m \in [0; h - 1]$, $|\mathcal{E}_m| \leq |\mathcal{E}_{m+1}|$.*

Démonstration. Nous allons montrer que pour tout $m \leq h - 1$, pour chaque état $E \in \mathcal{E}_m$, on peut construire un état $F(E) \in \mathcal{E}_{m+1}$, tel que pour tous $E_1 \in \mathcal{E}_m$ et $E_2 \in \mathcal{E}_m$, avec $E_1 \neq E_2$, on a $F(E_1) \neq F(E_2)$. Si cette propriété est vraie, on a bien $|\mathcal{E}_m| \leq |\mathcal{E}_{m+1}|$ (car la fonction F réalise alors une injection de \mathcal{E}_m dans \mathcal{E}_{m+1}).

Notons tout d'abord X le plus grand ensemble de nœuds tel que le graphe induit par X est un chemin ayant a comme extrémité. Notons P ce chemin. Soit $E \in \mathcal{E}_m$. Notons $x(E)$ le premier nœud du chemin P qui n'est pas dans E (ce nœud existe car $m \leq h - 1$). Posons $F(E) = E \cup \{x(E)\}$. On a bien $F(E) \in \mathcal{E}_{m+1}$.

Par l'absurde, prenons deux états E_1 et E_2 tels que $F(E_1) = F(E_2)$, et supposons que $E_1 \neq E_2$. On remarque que $x(E_1) \neq x(E_2)$ (sinon on déduirait de $F(E_1) = F(E_2)$ que $E_1 = E_2$, ce qui est impossible). Supposons sans perte de généralité que $x(E_1)$ est avant $x(E_2)$ sur le chemin P . On a donc $x(E_1) \in E_2$. Comme $|E_1| = |E_2|$ et $E_1 \neq E_2$, il existe un nœud y tel que $y \in E_1$, $y \notin E_2$. On sait que $y \neq x(E_1)$ car $x(E_1) \in E_2$, et on sait que $y \neq x(E_2)$ car $x(E_2) \notin E_1$. Ainsi, $F(E_1) = E_1 \cup \{x(E_1)\}$ contient y , car $y \in E_1$. Toutefois, $F(E_2) = E_2 \cup \{x(E_2)\}$ ne contient pas y , car $y \notin E_2$ et $y \neq x(E_2)$. On aboutit à $F(E_1) \neq F(E_2)$, ce qui est une contradiction, qui vient de l'hypothèse que $E_1 \neq E_2$. La propriété est donc vraie. \square

Graphes d'évolution pour des réseaux particuliers.

Dans la suite, nous donnons quelques exemples de graphes d'évolution obtenus lorsque la topologie du réseau est particulière.

Propriété 27. Si G est une étoile de n nœuds centrée sur la graine a , $\mathcal{T}(G, a)$ est isomorphe à un hypercube de dimension $n - 1$, à l'exception de l'état \emptyset .

Propriété 28. Si G est une clique de n nœuds et a la graine, $\mathcal{T}(G, a)$ est isomorphe à un hypercube de dimension $n - 1$, à l'exception de l'état \emptyset .

Propriété 29. Si G est un chemin de n nœuds et si la graine a est l'une des extrémités de ce chemin, $\mathcal{T}(G, a)$ est un chemin de $n + 1$ nœuds.

Propriété 30. Si G est un chemin de n nœuds et si la graine a est le m -ème nœud du chemin, $\mathcal{T}(G, a)$ est isomorphe à un pavé de dimension deux (dont les côtés sont de longueur m et $n - m + 1$), à l'exception de l'état \emptyset .

A.3 Compléments sur les sup-irréductibles

Propriété 31. Soit G un graphe, a la graine, d un nœud quelconque, et notons $P = (x_0 = a, x_1, \dots, x_k = d)$ le plus court chemin de a à d dans G . Alors, tous les états E_i de la forme $E_i = \{x_0, x_1, \dots, x_i\}$ sont des sup-irréductibles dans $\mathcal{T}(G, a)$.

Démonstration. Soit $P = (x_0 = a, x_1, \dots, x_n = d)$ un chemin de longueur n (avec $n \geq 1$) et $E_n = \{x_0, x_1, \dots, x_n\}$. L'état $E_{n-1} = \{x_0, x_1, \dots, x_{n-1}\}$ est un prédécesseur de l'état E_n . Nous devons montrer que E_n n'a pas d'autre prédécesseur que E_{n-1} . Supposons qu'il existe un autre prédécesseur à E_n , noté E' . On a donc $E_n = E' \cup \{x_n\}$, avec $x_n \notin E'$. Ainsi, $x_n \in E'$. Puisque x_n est couvert par E' , il faut qu'il existe un chemin de longueur n (ou moins) qui va de a à x_n . Or, nous avons supposé que P était le plus court chemin de a à d . Par conséquent, E' n'existe pas, et E_n n'a qu'un seul prédécesseur. Ainsi, E_n est bien un sup-irréductible. \square

Propriété 32. Le nombre de sup-irréductibles dans $\mathcal{T}(G, a)$ est supérieur ou égal à n (et il s'agit d'une borne inférieure).

Démonstration courte. Cela vient du fait que tous les nœuds de G sont sur un plus court chemin, et on peut donc leur faire correspondre un état sup-irréductible (au moins). De plus, quand G est un chemin et a une extrémité de ce chemin, cette borne est atteinte. \square

Le nombre de sup-irréductibles de $\mathcal{T}(G, a)$ est en $\mathcal{O}(2^n)$, avec n le nombre de nœuds de G . En effet, considérons le graphe G représenté sur la figure A.2. Le graphe possède $3k + 1$ nœuds, k étant le nombre de boucles. À chaque boucle, il y a deux chemins possibles. On peut donc compter au moins 2^k chemins différents qui mènent de a au nœud le plus à droite. Tous ces chemins correspondent à des états différents, et sont tous des sup-irréductibles. Le graphe possède donc au moins 2^k sup-irréductibles, alors que le graphe ne possède que $\mathcal{O}(k)$ nœuds. On a bien un nombre exponentiel de sup-irréductibles dans le pire des cas.

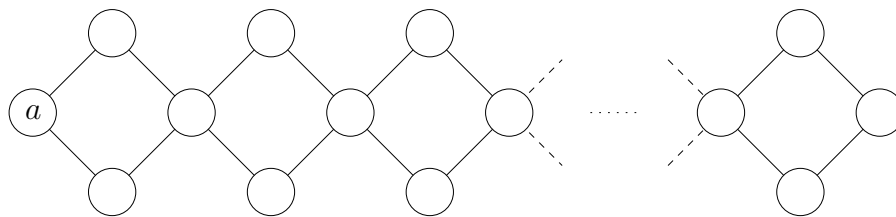


Figure A.2 – Un graphe G pour lequel le treillis $\mathcal{T}(G, a)$ possède un nombre exponentiel de sup-irréductibles.

ANNEXE

B

LEXIQUE DES ABRÉVIATIONS ET SIGLES UTILISÉS

ACM *Association for Computing Machinery*

AERES Agence d'Évaluation de la Recherche et de l'Enseignement Supérieur

AMI Appel à Manifestation d'Intérêt

ANR Agence Nationale de la Recherche

AODV *Ad-hoc On-demand Distance Vector*

APP Agence pour la Protection des Programmes

Armor Architectures et Modèles Réseaux

ASK *Amplitude-Shift Keying*

Assent *Ad-hoc Scalable Sensor Network*

Aurefi Association Universitaire Rouennaise des Étudiants de la Filière Informatique

BE *Backoff Exponent*

BI *Beacon Interval*

BO *Beacon Order*

BPSK *Binary Phase-Shift Keying*

BRGM Bureau de Recherches Géologiques et Minières

CAP *Contention Access Period*

CCA *Clear Channel Assessment*

CFP *Collision Free Period*

- Clervolc** Centre Clermontois de Recherche sur le Volcanisme
- CM** Cours magistral
- CNRS** Centre National de la Recherche Scientifique
- CNU** Conseil National des Universités
- CPAN** Coordinateur du PAN
- CSMA** *Carrier-Sense Multiple Access*
- CSMA/CA** *Carrier-Sense Multiple Access with Collision Avoidance*
- DCNS** Direction Technique des Constructions Navales
- DSSS** *Direct-Sequence Spread Spectrum*
- EDF R&D** Électricité de France - Recherche et Développement
- EPSRC** *Engineering and Physical Sciences Research Council*
- FFD** *Full Function Devices*
- GNU** *GNU is not Unix*
- GPLv3** *GNU General Public License version 3*
- GPS** *Global Positioning System*
- GTS** *Guaranteed Time Slot*
- IEEE** *Institute of Electrical and Electronics Engineers*
- IFIP** *International Federation for Information Processing*
- Inria** Institut National de Recherche en Informatique et en Automatique
- IPv4** *Internet Protocol version 4*
- IPv6** *Internet Protocol version 6*
- Irem** Institut de Recherche en Enseignement des Mathématiques
- Irisa** Institut de Recherche en Informatique et en Systèmes Aléatoires
- Irit** Institut de Recherche en Informatique de Toulouse
- ISM** *Industrial, Scientific and Medical radio bands*
- ISN** Informatique et Science du Numérique
- IUT** Institut Universitaire de Technologies
- L1** Licence première année
- L2** Licence deuxième année
- L3** Licence troisième année
- Labex** Laboratoire d'Excellence
- Lamp** Laboratoire de Météorologie Physique
- Lapsco** Laboratoire de Psychologie Sociale et Cognitive
- Lattis** Laboratoire Toulousain de Technologie et d'Ingénierie des Systèmes
- Limos** Laboratoire d'Informatique, de Modélisation, et d'Optimisation des Systèmes

Lirmm Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

LLC *Logical Link Control*

LM Laboratoire de Mathématiques de Clermont-Ferrand

LMV Laboratoire Magmas et Volcans

LNCS *Lecture Notes in Computer Science*

LNICST *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*

LPC Laboratoire de Physique Corpusculaire

LQI *Link Quality Indicator*

LRI Laboratoire de Recherche en Informatique

M2 Master deuxième année

MAC *Medium Access Control*

MPR *Multi-point Relay*

NS2 *Network Simulator version 2*

O-QPSK *Offset Quadrature Phase-Shift Keying*

Ocari Optimisation des Communications Ad-hoc pour les Réseaux Industriels

OLSR *Optimized Link State Routing protocol*

OPGC Observatoire de Physique du Globe de Clermont-Ferrand

OSI *Open Systems Interconnection*

PAN *Personal Area Network*

PES Préparation aux Études Scientifiques

PSSS *Parallel-Sequence Spread Spectrum*

RFD *Reduced Function Devices*

RTS/CTS *Request To Send / Clear To Send*

SCP *Set Cover Problem*

SD *Superframe Duration*

SO *Superframe Order*

SP5 Sous-projet 5

SPI *Serial Peripheral Interface*

SQL *Structured Query Language*

TD Travaux dirigés

TDMA *Time-Division Multiple Access*

TIME-EACM *Transport Information Monitoring Environment - Event Architecture and Context Management*

TP Travaux pratiques

UFR Unité de Formation et de Recherche