HAL
open science

Réseaux cognitifs sans fil pour des applications industrielles 4.0

Alexis Bitaillou

To cite this version:

Alexis Bitaillou. Réseaux cognitifs sans fil pour des applications industrielles 4.0. Informatique [cs]. Université de Nantes (FR), 2021. Français. NNT : 2021NANT4077 . tel-03638680

HAL Id: tel-03638680

https://theses.hal.science/tel-03638680

Submitted on 12 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE NANTES

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE NANTES

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Alexis BITAILLOU

Réseaux cognitifs sans fil pour des applications industrielles 4.0

Thèse présentée et soutenue à Nantes, le 15 novembre 2021
Unité de recherche : Laboratoire des Sciences de Numérique de Nantes (LS2N), UMR 6004

Rapporteurs avant soutenance :

Fabrice THEOLEYRE	Chargé de recherche CNRS, CNRS/Université de Strasbourg
Fabrice VALOIS	Professeur des universités, INSA Lyon

Composition du Jury :

Président :	Nicolas MONTAVONT	Professeur de l'Ecole des Mines, IMT Atlantique
Examinateurs :	Fabrice THEOLEYRE	Chargé de recherche CNRS, CNRS/Université de Strasbourg
	Fabrice VALOIS	Professeur des universités, INSA Lyon
	Nathalie MITTON	Directeur de recherche, INRIA Lille-Nord Europe
Dir. de thèse :	Benoît PARREIN	Maître de Conférence HDR, Université de Nantes
Co-encadrant. de thèse :	Guillaume ANDRIEUX	Maître de Conférence HDR, Université de Nantes

Invité(s) :

Dominique BARTHEL	Principal Research Engineer, Orange Labs Grenoble

Remerciements

Ce manuscrit est le fruit de mes trois années de doctorat. Il est d'abord le résultat de ma scolarité. Je remercie tous les enseignants qui ont tenté de m'apprendre et me faire comprendre leur connaissance et leur savoir. Sans leur professionnalisme et leur compétence, je ne pense pas que j'aurais été aussi loin.

Je remercie Nicolas MONTAVONT, Nathalie MITTON et Dominique BARTHEL d'avoir accepté de participer au jury. Je remercie particulièrement Fabrice THEOLEYRE et Fabrice VALOIS pour avoir rapporté cette thèse. Je remercie l'ensemble du jury d'avoir lu attentivement ce manuscrit afin de trouver les questions les plus intéressantes possibles pendant la soutenance.

Je remercie mes encadrants Benoît PARREIN et Guillaume ANDRIEUX pour m'avoir encadré et aidé durant ces trois années. Je les remercie de leur soutien sans faille et leur disponibilité que ce soit pendant les succès ou les moments plus compliqués comme le premier confinement.

Je remercie ma famille qui m'a soutenu pendant ma scolarité et pendant mon doctorat.

Enfin, je remercie les RFI WISE et RFI Atlanstic 2020 pour leur soutien financier. Leur soutien a permis que mon doctorat se passe dans les meilleures conditions matérielles possibles.

Table des matières

Introduction générale **7**

I Contexte et état de l'art **9**

Introduction de la première partie **11**

1 L'Industrie du futur 4.0 **13**
1.1 Introduction 13
1.2 Les origines de l'Industrie 4.0 14
1.3 Les ambitions de l'Industrie 4.0 15
1.3.1 Une optimisation des moyens et des ressources 15
1.3.2 Vers plus de flexibilité 16
1.3.3 Des technologies standardisées et ouvertes 16
1.3.4 Sécurité informatique 17
1.4 Conclusions 17

2 Réseaux de communication pour l'industrie du futur **19**
2.1 Introduction 19
2.2 Réseaux câblés 20
2.2.1 Communication série 20
2.2.2 Ethernet et ses dérivés 21
2.2.3 Conclusions 24
2.3 Wireless Personal Area Network (WPAN) 24
2.3.1 IEEE 802.15.4 24
2.3.2 WIA-PA 25
2.3.3 ISA100.11a 25
2.3.4 WirelessHART 26
2.3.5 Conclusions 26
2.4 Wireless Local Area Network 26
2.4.1 IEEE 802.11 27

2.5 Conclusions 28

3 Vers des réseaux intelligents 31
3.1 Introduction 32
3.2 Les problèmes d'interférences radio 32
3.2.1 Interférences multi-trajets 32
3.2.2 Interférences multi-utilisateurs 33
3.2.3 Gestion des collisions 33
3.2.4 Protocole de routage optimisé 33
3.2.5 Concurrence inter-protocoles 34
3.3 Optimisation inter-couches 37
3.3.1 Le principe de l'optimisation inter-couches 37
3.3.2 Exemples d'application 39
3.3.3 Les limites de l'optimisation inter-couches 42
3.4 Réseaux cognitifs 42
3.4.1 Définition d'un réseau cognitif 43
3.4.2 Quelques exemples d'applications des réseaux cognitifs 43
3.4.3 Architecture, implémentations et résultats 44
3.4.4 Conclusions sur les réseaux cognitifs 50
3.5 Conclusions 51

4 L'apprentissage par renforcement appliqué au routage 53
4.1 Introduction 53
4.2 Apprentissage par renforcement 53
4.2.1 Quelques exemples d'algorithmes d'apprentissage par renforcement . . . 55
4.3 Applications de l'apprentissage par renforcement aux problèmes de réseaux informatiques 58
4.3.1 Application au paramétrage automatique du protocole TCP 58
4.3.2 Application à la fonction de routage 59
4.4 Q-routing 60
4.4.1 Fonctionnement de Q-routing 60
4.4.2 Analyse des résultats obtenus par Boyan et Littman 61
4.5 Modifications génériques de Q-routing 63
4.5.1 Predictive Q-routing 63
4.5.2 Dual Reinforcement Q-Routing 63
4.5.3 Q-Neural Routing 64
4.5.4 K-Shortest path Q-routing 64
4.5.5 Q^2-routing 64
4.6 Adaptation de Q-routing à des scénarios particuliers 65
4.6.1 Q-routing sur des scénarios avec mobilité 65
4.6.2 Q-routing pour les réseaux de capteurs 66

4.6.3 Q-routing pour la radio cognitive . . . 66
4.7 Conclusions . . . 66

Conclusion de la première partie **69**

II Contributions **71**

Introduction de la deuxième partie **73**

5 Q-routing : de l'algorithme au protocole **75**
5.1 Introduction . . . 75
5.2 Implémentations de Q-routing . . . 76
5.2.1 Mesure du délai et format des entêtes . . . 76
5.2.2 Cycle de vie des routes . . . 77
5.2.3 Choix du taux d'apprentissage η . . . 78
5.3 Évaluation de l'implémentation . . . 78
5.3.1 Exemple jouet . . . 78
5.3.2 La grille irrégulière de Boyan et Littman . . . 81
5.4 Conclusions . . . 85

6 Atténuation des effets de la stratégie gloutonne de Q-routing **87**
6.1 Introduction . . . 87
6.2 Mise en évidence des effets de la stratégie gloutonne . . . 88
6.3 Solutions proposées . . . 90
6.3.1 Approche probabiliste inspirée de l'apprentissage par renforcement . . . 90
6.3.2 Système d'époque . . . 91
6.4 Évaluation des solutions proposées . . . 92
6.4.1 Exemple jouet sans-fil . . . 92
6.4.2 Grille sans-fil inspirée de Boyan et Littman . . . 96
6.5 Conclusions . . . 103

7 Amélioration de la mesure du délai de Q-routing **105**
7.1 Introduction . . . 105
7.2 Mise en évidence des problèmes d'estimation . . . 106
7.2.1 Présentation du problème . . . 106
7.2.2 Scénario . . . 106
7.2.3 Résultats . . . 107
7.3 Solutions proposées . . . 109
7.3.1 Filtrage des mesures de délais . . . 109
7.3.2 Route à trois statuts . . . 109
7.4 Évaluation des solutions proposées . . . 109

7.4.1 Scénario avec mobilité 110
7.4.2 Grille sans-fil inspirée de Boyan et Littman 113
7.5 Conclusions 115

Conclusion de la deuxième partie **119**

Conclusion générale **121**

Perspectives **125**

Production scientifique **127**

Productions logicielles **129**

Références bibliographiques **131**

Table des figures **149**

Introduction générale

La notion d'Industrie 4.0 est née en 2011 en Allemagne. Elle traduit la volonté d'augmenter la compétitivité de l'industrie allemande face à la concurrence d'autres nations. Pour cela, l'Industrie 4.0 propose des objectifs techniques et organisationnels ambitieux comme une amélioration de la flexibilité et une meilleure gestion des moyens de production. Cela implique par exemple l'utilisation de technologies ouvertes et standardisées comme celle de l'IEEE (*Institute of Electrical and Electronics Engineers*) ou de l'IETF (*Internet Engineering Task Force*). L'Industrie 4.0 se démarque par l'utilisation de l'Internet des Objets et des systèmes cyber-physiques (*Cyber-Physical Systems*). Un système cyber-physique est un système connecté à un réseau d'information et interagissant avec le monde physique. Les robots et les capteurs sont des exemples de systèmes cyber-physiques. Pour relier ces systèmes, les réseaux sans-fil sont une solution efficiente maximisant la flexibilité. Malheureusement, les réseaux sans-fil sont sensibles aux phénomènes d'interférences. Pour y remédier, une solution au problème d'interférences peut être la radio cognitive. La radio cognitive consiste à reparamétrer automatiquement les paramètres des couches basses (physique, voire liaison) afin d'augmenter entre autres le débit ou la fiabilité. Ce concept a été étendu dans les réseaux cognitifs. Les réseaux cognitifs sont des réseaux capables d'optimiser automatiquement les différents paramètres des couches de services afin d'atteindre un ou plusieurs objectifs de performances comme l'amélioration de la fiabilité ou la diminution du délai. Les réseaux cognitifs peuvent répondre aux besoins de fiabilité et de qualité de service des réseaux industriels selon nous. L'objectif de cette thèse est d'étudier les réseaux cognitifs et de les intégrer dans les réseaux industriels sans-fil. Cette thèse s'inscrit dans le cadre du projet COWIN (COgnitive Wireless networks for Industrial applicatioNs) soutenu par les RFI Wise et Atlanstic 2020, qui sont deux programmes Recherche, Formation et Innovation de la région Pays de la Loire.

La première partie de cette thèse porte sur un état de l'art. Dans le Chapitre 1, nous commençons par présenter l'Industrie 4.0 qui fixe des objectifs techniques et organisationnelles afin d'accroître la compétitivité de l'industrie. Contrairement aux précédentes révolutions industrielles, l'objectif n'est pas d'augmenter la capacité de production, mais d'améliorer la flexibilité et la gestion des ressources. Pour cela, l'Industrie 4.0 s'inspire notamment du *cloud computing* en favorisant l'intégration verticale et utilisant une gestion dynamique et à la demande des différentes ressources. Comme le *cloud computing*, l'Industrie 4.0 repose en partie sur des technologies ouvertes et standardisées notamment celles issues de l'IEEE et de l'IETF.

Dans le Chapitre 2, nous abordons ensuite les technologies des réseaux industriels. Elles sont

pour la plupart propriétaires et certaines non-compatibles avec les réseaux informatiques. Cependant, des technologies des réseaux informatiques sont maintenant capables de soutenir certaines contraintes des milieux industriels. De plus, elles ont l'avantage d'être ouvertes et standardisées.

Dans le Chapitre 3, nous présentons d'abord les problématiques liées aux interférences radio. En effet, la plupart des réseaux sans-fil utilisent les mêmes bandes de fréquences. Ce qui favorise l'apparition d'interférences. Ces interférences détruisent les trames et donc les données. Dans un second temps, nous nous intéressons à l'optimisation inter-couches. Les optimisations inter-couches reposent sur la communication entre des couches de services potentiellement non adjacentes pour améliorer les performances. Enfin, nous terminons ce chapitre avec les réseaux cognitifs. Les réseaux cognitifs sont des réseaux capables d'optimiser les paramètres de l'ensemble des couches du modèle OSI. Néanmoins, les réseaux cognitifs sont si complexes à mettre en œuvre que la majorité des travaux se sont arrêtés aux spécifications. Face à cette complexité, la plupart des auteurs ont préféré se concentrer sur l'amélioration d'une couche protocolaire. Les résultats obtenus grâce à ces améliorations nous invite à poursuivre dans cette voie.

À cette fin, nous nous attachons dans le Chapitre 4 à l'utilisation de l'apprentissage par renforcement appliquée aux réseaux informatiques. L'apprentissage par renforcement est l'un des trois paradigmes de l'apprentissage automatique. Sa particularité est d'apprendre par essais et erreurs. Il est donc adapté aux environnements dynamiques que sont les réseaux informatiques. De plus, l'apprentissage par renforcement ne nécessite pas nécessairement de jeu de données initial. Les algorithmes présentés appartiennent tous à la catégorie des processus de décision Markovien (MDP). Parmi tous les algorithmes incluant de l'apprentissage par renforcement, Q-routing [BL94] nous a paru le plus prometteur par ses résultats et ses travaux dérivés. Q-routing est un algorithme de routage inspiré de Q-learning, un algorithme d'apprentissage par renforcement.

La seconde partie du manuscrit détaille nos contributions. Le Chapitre 5 présente deux de nos contributions. La première contribution est l'implémentation de Q-routing sur le simulateur réseaux Qualnet. Nous détaillons notamment les caractéristiques et nos choix d'implémentation. La seconde contribution présente dans ce chapitre est l'évaluation de notre implémentation sur la grille irrégulière de Boyan et Littman [BL94], les auteurs de Q-routing.

Dans le Chapitre 6, pour notre troisième contribution, nous proposons deux solutions inspirées de l'apprentissage par renforcement pour atténuer un effet de la stratégie gloutonne de Q-routing. Nous les évaluons et nous montrons que la réinitialisation périodique est la solution la plus efficace. Q-routing a été conçu et évalué sur les réseaux filaires uniquement. La quatrième contribution réside dans l'évaluation de Q-routing sur deux topologies sans-fil incluant une version adaptée de la grille de Boyan et Littman.

Dans le Chapitre 7, nous mettons en évidence que la métrique de routage de Q-routing peut être bruitée à cause d'une mauvaise estimation du délai. Dans notre cinquième contribution, nous émettons deux propositions pour résoudre ce problème : *i)* appliquer une méthode de filtrage afin d'améliorer l'estimation ou *ii)* utiliser une valeur prédite à partir de plusieurs mesures antérieures. Le filtrage s'avère être la seule proposition efficace. Pour notre dernière contribution, nous proposons une première évaluation de notre implémentation de Q-routing dans un scénario avec mobilité.

partie I

Contexte et état de l’art

Introduction de la première partie

Au début des années 2010, le ministère de l'industrie allemand commande à un groupe d'experts un rapport sur comment sécuriser l'avenir de son industrie. En effet, à ce moment, l'industrie allemande subit la pression de la concurrence d'autres industries. En 2013, ce groupe d'experts livre le rapport contenant leurs solutions pour augmenter la compétitivité de l'industrie allemande [Kag13]. Il propose l'Industrie 4.0.

Dans le cadre de l'Industrie 4.0, le rapport définit des objectifs technologiques et organisationnels afin de répondre aux défis des industries concurrentes. Dans le premier chapitre, nous détaillons les ambitions et quelques objectifs choisis de l'Industrie 4.0. Parmi ces objectifs technologiques, il y a l'utilisation d'un ensemble de technologies ouvertes et standardisées.

Concernant les réseaux industriels, il existe déjà un grand nombre de technologies [WSJ17]. Certaines d'entre elles comme les bus de terrain ne sont pas compatibles avec les réseaux informatiques. D'autres comme la plupart des solutions dérivées d'Ethernet sont propriétaires. Pour répondre à ce nouveau besoin, deux technologies IEEE ont évolué. Il s'agit d'Ethernet pour les réseaux filaires et IEEE 802.15.4 pour les réseaux sans-fil. Dans le Chapitre 2, nous abordons notamment les technologies des réseaux industriels dérivées d'Ethernet et celles dérivées d'IEEE 802.15.4.

La multiplication des réseaux sans-fil engendre la saturation du spectre radio. Les réseaux sans-fil utilisent majoritairement les même bandes de fréquences ISM (Industrielle, Scientifique et Médicale) et notamment les bandes ISM 868 MHz et 2,4 GHz. Cette (sur-)utilisation de ces bandes ISM entraîne une mise en concurrence protocolaire. Une solution à ce problème est la radio cognitive. Un des rôles de la radio cognitive consiste à améliorer automatiquement les paramètres radio des réseaux sans-fil. Cette fonctionnalité a inspiré les réseaux cognitifs. Les réseaux cognitifs sont des réseaux capables d'optimiser automatiquement les paramètres des différentes couches protocolaires afin d'atteindre un ou plusieurs objectifs de performances. Le Chapitre 3 sera consacré principalement aux réseaux cognitifs.

Malheureusement, les réseaux cognitifs ont été très peu mis en œuvre à cause de leur complexité. Les auteurs ont préféré se focaliser sur l'amélioration d'une couche de services, généralement grâce à l'apprentissage par renforcement. Dans ce dernier chapitre, nous abordons ce troisième paradigme de l'apprentissage automatique à travers quelques algorithmes de la famille de Q-learning ainsi que quelques applications. Nous nous focaliserons sur Q-routing, un algorithme de routage inspiré de l'algorithme d'apprentissage par renforcement Q-learning. Nous verrons aussi quelques travaux d'améliorations de Q-routing.

Chapitre 1

L'Industrie du futur 4.0

Sommaire

2.1 Introduction **19**
2.2 Réseaux câblés **20**
2.2.1 Communication série 20
2.2.2 Ethernet et ses dérivés 21
2.2.3 Conclusions 24
2.3 Wireless Personal Area Network (WPAN) **24**
2.3.1 IEEE 802.15.4 24
2.3.2 WIA-PA 25
2.3.3 ISA100.11a 25
2.3.4 WirelessHART 26
2.3.5 Conclusions 26
2.4 Wireless Local Area Network **26**
2.4.1 IEEE 802.11 27
2.5 Conclusions **28**

1.1 Introduction

Avec le perfectionnement de l'électronique et le développement des nouvelles technologies de l'information et de la communication (NTIC) à la fin du XXe siècle, l'industrie entra dans une nouvelle ère. L'industrie 3.0 commence dans le début des années 70. Elle renforce l'automatisation et introduit l'informatique dans l'industrie. L'objectif était que les ouvriers n'aient plus à faire les tâches simples et répétitives. Cela implique une réduction de la masse salariale non qualifiée, et aussi l'emploi de personnel qualifié pour la maintenance des équipements. En complément de l'automatisation, l'informatique dans l'entreprise permet une meilleure gestion des ressources. Par

exemple, les entreprises s'équipent de progiciel de gestion intégré, ce qui permet la centralisation des informations et de la planification. Les communications sont aussi facilitées par le déploiement de réseaux haut-débit et de l'Internet. Mais l'industrie poursuit sa course à la compétitivité. La Figure 1.1 montre les différentes révolutions industrielles. Elle montre finalement que l'industrie doit adopter les nouvelles technologies de son époque pour rester compétitive. De nos jours, l'informatique et l'automatisation ne sont plus des nouvelles technologies. Elles ne sont plus suffisantes pour se différentier. Cela implique que l'industrie 3.0 n'est plus suffisante pour qu'une industrie soit compétitive. Il faut donc de nouvelles normes et technologies pour l'industrie. C'est là qu'intervient l'Industrie 4.0. Dans ce même chapitre, nous commencerons par aborder les origines de l'Industrie 4.0. Puis, nous terminerons par les ambitions et les nouvelles contraintes de l'Industrie 4.0.

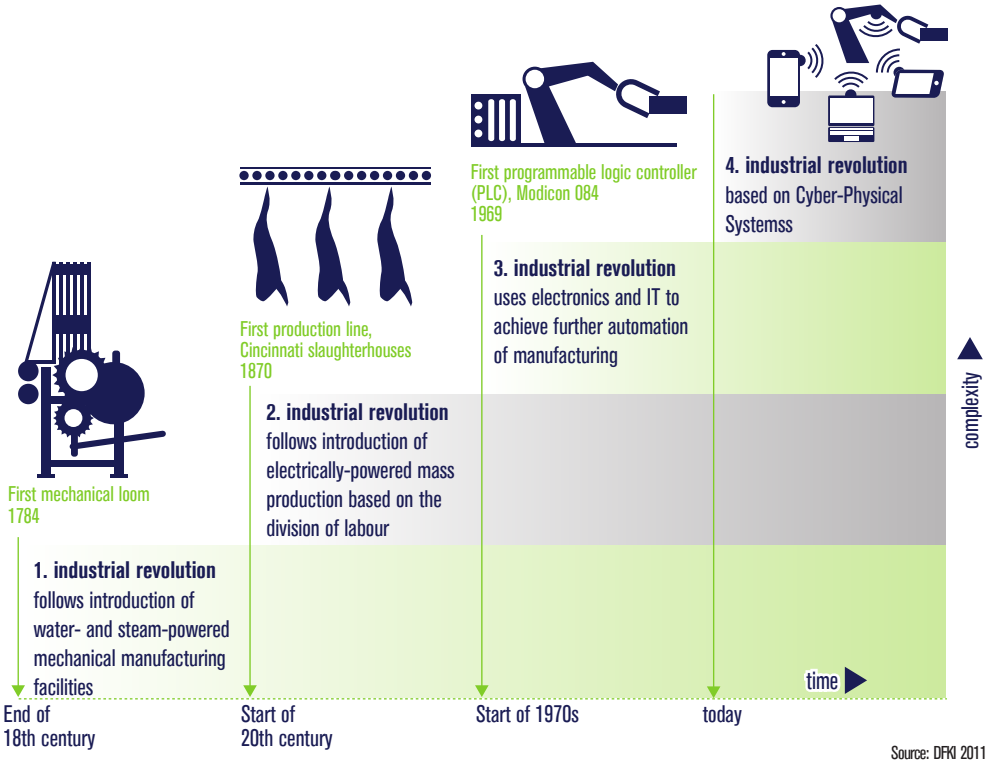


FIG. 1.1 : De la première révolution industrielle à l'Industrie 4.0, *extrait du site du Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)*

1.2 Les origines de l'Industrie 4.0

L'Industrie 4.0 possède quelques prérequis techniques. Avant de passer à l'Industrie 4.0, il faut déjà avoir la maîtrise de l'industrie 3.0 [Chr17]. Comme il était indispensable d'avoir l'électrisation

pour utiliser la robotique, il est indispensable d'utiliser l'automatisation et les NTIC pour commencer à mettre en œuvre l'Industrie 4.0. L'autre prérequis concerne l'Internet des Objets (*Internet of Things*, IoT). L'IoT est un domaine qui est apparu durant la dernière décennie. Il repose surtout sur les capteurs et les terminaux de faible puissance connectés à Internet. L'Industrie 4.0 repose aussi sur les systèmes cyber-physiques. Un système cyber-physique est un système où des éléments informatiques contrôlent les éléments physiques. Un distributeur de billet de banque est un exemple de système cyber-physique car c'est un système connecté à un réseau informatique et manipulant des objets (les billets de banque). L'Internet des objets adapté à l'industrie et les systèmes cyber-physiques sont considérés comme des éléments fondamentaux de l'Industrie 4.0 [ZLZ15].

Les origines de l'Industrie 4.0 sont relativement simples. Le concept d'Industrie 4.0 est développé dans une étude sponsorisée par le ministère fédéral allemand de l'enseignement et de la recherche. Il a été demandé à un comité composé de chercheurs académiques et industriels de se pencher sur la question « comment sécuriser l'avenir de l'industrie manufacturière allemande » [Kag13]. La réponse de ce comité est un rapport d'environ 80 pages publié en 2013. Il fait rapidement un état des lieux de l'industrie au niveau mondial et propose un ensemble de recommandations pour mettre en place l'Industrie 4.0. L'industrie allemande fait partie des leaders mondiaux. Néanmoins, elle est menacée par l'industrie chinoise et étasunienne. L'industrie chinoise est capable de produire des biens en grande quantité et à faible coût grâce à sa capacité de production importante et ses coûts salariaux plus faibles. Ce n'est pas le cas de l'industrie allemande. Elle doit donc trouver un moyen de rester compétitive, voire se démarquer des autres industries. Un autre élément vient handicaper l'industrie allemande. Le nombre de personnes hautement qualifiées est insuffisant pour satisfaire la demande. Le déficit tend même à se creuser dans les prochaines années. À défaut de pouvoir recruter, l'industrie allemande doit soit garder ses employés plus longtemps actifs, soit les faire évoluer en interne.

Il existe une initiative française nommée « Nouvelle France Industrielle » (NFI) depuis 2015 [Bel15]. Elle s'appuie aussi sur l'industrie du futur. Elle définit neuf secteurs clés : l'économie des données, les objets intelligents, la confiance numérique, l'alimentation intelligente, les nouvelles ressources, la ville durable, la mobilité écologique, les transports de demain et la médecine du futur. Enfin, ce projet précise 10 solutions dont l'usine du futur, les objets intelligents et la confiance numérique.

1.3 Les ambitions de l'Industrie 4.0

L'objectif principal de l'Industrie 4.0 est de créer de la valeur ajoutée dans l'industrie manufacturière, indépendamment des coûts salariaux [Kag13]. Il est possible de découper cet objectif en plusieurs sous-objectifs : l'optimisation des ressources et des moyens, une flexibilité accrue, une standardisation des technologies déployées et la sécurité informatique.

1.3.1 Une optimisation des moyens et des ressources

Le premier point est d'optimiser le processus de fabrication. Pour cela, il faut que les équipements de production soient le plus polyvalent possible. L'objectif est de pouvoir remplacer une

machine par une autre en cas de besoin. Par exemple, en cas de maintenance ou en cas de changement du bien à produire, l'usine ne doit pas être dépendante d'une machine. De plus, la chaîne de production doit être flexible. Elle doit pouvoir s'adapter rapidement au processus de production.

Une chaîne de production capable de s'adapter rapidement permet d'éviter les temps morts. Ce n'est qu'un élément pour optimiser les coûts. D'autres éléments comme la gestion des matières premières sont aussi importants. La nouveauté de l'Industrie 4.0 est la possibilité de simuler le niveau des stocks, le transport et la logistique. Cela permettra de mieux anticiper les approvisionnements par exemple. Les matières premières ne sont pas les seules ressources utilisées par l'industrie, il y a aussi l'énergie. La gestion de l'énergie peut être améliorée notamment pendant les week-ends et les périodes de vacances. L'idée de Siemens repris dans [Kag13] est de pouvoir passer l'usine en « mode veille » à la manière d'un simple ordinateur ou du *cloud computing*. L'efficacité et l'efficience des ressources utilisées est un point clé pour réduire le coût des produits.

1.3.2 Vers plus de flexibilité

La flexibilité des processus de fabrication est un élément clé de l'Industrie 4.0. L'Industrie 4.0 permet la création de valeur par la personnalisation du bien par le client. L'objectif est que chaque client puisse demander un produit sur-mesure. Pour cela, il faut donc pouvoir adapter le processus de fabrication en fonction des besoins. Les équipements doivent être reconfigurables « à la volée ». Selon Grier [Gri17], l'idée est que les usines deviennent reconfigurables rapidement et à très faible coût à la manière du *cloud computing*. Ce qui implique le développement de solutions équivalentes à celles présentes dans le *cloud computing*. Nous retrouvons ce lien avec le *cloud computing* dans le nom de certains modèles [Xu12]. Par exemple, un modèle d'implémentation de l'Industrie 4.0 porte le nom de *cloud manufacturing*. Comme le *cloud computing*, l'Industrie 4.0 doit supporter l'intégration de bout-en-bout de la chaîne de valeur. C'est-à-dire qu'il faut que tous les éléments de la chaîne, des fournisseurs aux clients finaux en passant par le fabricant doivent être vus comme un ensemble, et plus comme des maillons indépendants. L'Industrie 4.0 doit aussi permettre l'intégration verticale. L'intégration verticale consiste à augmenter la capacité des éléments existants. L'intégration verticale d'une usine consiste à pouvoir ajouter une ligne de production sans construire une autre usine, par exemple. L'intégration horizontale consisterait à construire une usine supplémentaire de capacité comparable. Par analogie avec le *cloud computing*, l'intégration verticale consiste à augmenter la capacité d'un/des serveur(s). En intégration horizontale, un serveur supplémentaire est ajouté au groupe de serveurs.

1.3.3 Des technologies standardisées et ouvertes

Une recommandation pour l'Industrie 4.0 est de se reposer sur un unique ensemble de standards ouverts. L'exemple cité dans [Kag13] est l'ensemble des normes IEEE (*Institute of Electrical and Electronics Engineers*) et IETF (*Internet Engineering Task Force*). Cette recommandation est importante car elle permettrait aux entreprises de ne pas être dépendant technologiquement. Par exemple, des solutions commerciales comme Sercos ou PROFINET se tournent vers des standards

IEEE (Ethernet notamment) et IETF pour assurer leur pérennité. Malheureusement, une partie des technologies pour les réseaux industriels sont propriétaires. L'un des points bloquants est pour le moment la gestion du temps réel. Par exemple, Ethernet n'est pas particulièrement adapté au temps réel. Des solutions propriétaires comme les deux premières versions de Sercos ou PROFIBUS permettent une gestion « native » du temps réel et de la latence.

1.3.4 Sécurité informatique

Aux débuts de l'intégration des NTIC, la sécurité n'était pas une priorité. Les fonctionnalités ont été préférées à la sécurité. Les premiers virus comme le ver Morris [Orm03] ont montré que la sécurité ne devait plus être négligée. Ces menaces s'attaquent principalement aux systèmes informatiques des entreprises. Le but est soit de voler des données stratégiques, soit de paralyser l'entreprise. WannaCry [Ehr17] est un exemple du second cas. Ce rançongiciel est capable de paralyser l'intégralité d'un réseau informatique jusqu'à paiement d'une rançon. Des entreprises comme Saint-Gobain et Renault ont vu leurs parcs informatiques bloqués pendant plusieurs jours suites aux infections de rançongiciels. Un autre rançongiciel Petya et ses variantes auraient coûté plus d'un milliard d'euros aux entreprises selon une estimation du Monde [Ute17]. Les virus informatiques peuvent attaquer aussi les équipements industriels reliés aux réseaux informatiques. Stuxnet est un exemple d'attaque contre des équipements industriels à travers un réseau informatique [CA11 ; Kus13]. Stuxnet est un virus étatique destiné à freiner le programme nucléaire iranien. Il a commencé par infecter un grand nombre d'ordinateurs. Une fois sur le réseau de contrôle des équipements de la centrale nucléaire, le virus a pris possession du système de contrôle et d'acquisition de données. Enfin, il a faussé les informations et a fait fonctionner les centrifugeuses anormalement jusqu'à leur destruction. Stuxnet a démontré qu'il était possible de détruire des équipements physiques connectés à un système informatique. Dans un cadre industriel, les cyber-menaces peuvent bloquer le fonctionnement d'une entreprise et détruire des équipements, d'après ces exemples. Dans l'Industrie 4.0, la sécurité est donc considérée comme cruciale, au même titre que la sûreté. Pour les systèmes cyber-physiques, la sécurité doit être intégrée dès la conception (*security by design*).

1.4 Conclusions

L'industrie du futur n'est pas une révolution en soi. Elle s'appuie largement sur l'industrie 3.0. Néanmoins, elle apporte de nouvelles exigences technologiques et organisationnelles. La particularité de l'industrie du futur reste l'intégration de l'Internet des Objets et des systèmes cyber-physiques. Elle nécessite aussi une grande flexibilité que ce soit dans les processus de fabrication que dans les systèmes d'information notamment via l'intégration horizontale. L'Industrie 4.0 gagne en efficience en termes de ressources pour des raisons économiques. Cependant, le développement durable et la défense de l'environnement ne sont pas abordés. Enfin, l'industrie du futur repose notamment sur des technologies ouvertes et standardisées. Nous proposons d'ajouter de l'intelligence dans les réseaux informatiques afin de répondre aux besoins de fiabilité et de flexibilité

des réseaux. Dans le Chapitre 2, nous allons voir quelques technologies industrielles historiques et celles qui sont compatibles avec l'Industrie 4.0.

Chapitre 2

Réseaux de communication pour l'industrie du futur

Sommaire

3.1 Introduction **32**
3.2 Les problèmes d'interférences radio **32**
3.2.1 Interférences multi-trajets 32
3.2.2 Interférences multi-utilisateurs 33
3.2.3 Gestion des collisions 33
3.2.4 Protocole de routage optimisé 33
3.2.5 Concurrence inter-protocoles 34
3.3 Optimisation inter-couches **37**
3.3.1 Le principe de l'optimisation inter-couches 37
3.3.2 Exemples d'application 39
3.3.3 Les limites de l'optimisation inter-couches 42
3.4 Réseaux cognitifs **42**
3.4.1 Définition d'un réseau cognitif 43
3.4.2 Quelques exemples d'applications des réseaux cognitifs 43
3.4.3 Architecture, implémentations et résultats 44
3.4.4 Conclusions sur les réseaux cognitifs 50
3.5 Conclusions **51**

2.1 Introduction

Historiquement, les réseaux industriels sont composés d'éléments câblés. Les technologies comme Ethernet et RS232 sont éprouvées par des années d'utilisation. Mais l'utilisation de câbles implique

la faible mobilité des éléments connectés et un coût de redéploiement potentiellement élevé. Cette mobilité réduite commence à devenir problématique au commencement de l'ère de l'Industrie 4.0 [VT14]. Dans ce chapitre, nous allons commencer par les technologies câblées car elles sont historiquement présentes. Ensuite, nous verrons différentes technologies sans-fil développées pour remplacer les réseaux câblés. Nous ne pouvons pas être exhaustifs, car il existe un très grand nombre de technologies [WSJ17 ; Zur15]. Nous allons en voir quelques-unes choisies soit pour leur aspect historique, soit parce qu'elles sont liées à un standard IEEE.

2.2 Réseaux câblés

Les réseaux câblés ont la réputation d'être fiables, robustes, sûrs et performants, même s'ils ont quelques inconvénients comme les contraintes de déploiement ou la très faible mobilité des éléments connectés. En termes de débit et de fiabilité, les réseaux câblés sont encore, pour le moment, plus performants que les réseaux sans fil. Historiquement, ce sont la boucle de courant et RS-232 qui assuraient la communication entre un contrôleur et un actionneur ou un capteur. Ces solutions ne sont pas parfaites. Elles imposent des restrictions concernant les changements de topologie par exemple. Pour résoudre ces problèmes, des alternatives ont été développées. Comme elles sont orientées industries, elles sont appelées « bus de terrain ». Nous pouvons citer par exemple l'initiative française FIP (*Factory Instrumentation Protocol*) et allemande PROFIBUS. La standardisation des bus de terrain s'est révélée assez problématique [FS02]. La version initiale de l'IEC 61158 regroupe donc 8 technologies différentes. La version amendée de 2008 compte finalement 16 technologies [Dec09]. Il existe donc une grande quantité de solutions, l'article [Fel05] et l'ouvrage conséquent [Zur15] en dénombrent et en détaillent une partie. Nous avons choisi les communications séries pour leur aspect historique ainsi que les technologies dérivées d'IEEE 802.3.

2.2.1 Communication série

RS-232 (ou EIA 232) est une norme de communication de type série. Cette norme a été validée par l'EIA en 1962. TIA/EIA-232-F est la dernière révision parue en 1997. Elle permet un échange asynchrone et bidirectionnel entre deux équipements (point à point). La norme définit seulement le fonctionnement électrique de la connexion, le brochage et le connecteur. Les codages et le débit de transmission ne sont donc pas définis dans la norme. Par ses spécifications, RS-232 est donc une norme de couche physique dans la pile OSI. Par exemple, Il est donc possible d'utiliser PPP et IP sur RS-232. En pratique, UART (*Universal Asynchronous Receiver Transmitter*) est généralement utilisé sur RS-232. RS-232 peut atteindre un débit maximal de 115200 bit/s sur des très courtes distances. Dans son livre *PC 97 Hardware Design Guide* [Cor97], Microsoft déprécie l'utilisation de la norme RS-232 au profit de normes comme USB.

RS-485 (ou TIA/EIA 485) est une amélioration significative de TIA/EIA 422. TIA/EIA 422 est une extension de RS-232. Comme pour RS-232, la norme ne définit que le niveau physique, c'est-à-dire le fonctionnement électrique, le brochage et le connecteur. RS-485 supporte le point à point ainsi que le multipoint. Il est donc possible d'avoir une topologie de type bus via RS-485. Les connexions

peuvent aussi être établies en *full duplex*. En pratique, RS-485 peut atteindre un débit maximal de 10 Mbit/s sur de très courtes distances ou une distance de 1200 m avec un débit moindre [Max06].

2.2.2 Ethernet et ses dérivés

Ethernet

Ethernet est un protocole réseau développé par Xerox et publié en 1980 [SH80]. Ethernet a été standardisé par l'IEEE en 1983 sous le nom d'IEEE 802.3. Ethernet définit à la fois la couche physique et la couche liaison. En 1983, la première implémentation standardisée permettait un débit de 10 Mbit/s sur un câble coaxial [Sho+85]. En 1987, la paire cuivrée et la fibre optique furent ajoutées au standard. La dernière révision (802.3bs-2017) permet un débit maximal de 400 Gbit/s. Les premières implémentations ne permettaient que des topologies de type bus. Maintenant, Ethernet supporte les topologies en étoile et point à point. Concernant la partie liaison, les données transitent sous forme de trames. Les terminaux possèdent une adresse sur 48 bits, dépendante du constructeur [DP81]. La Figure 2.1 représente la structure d'une trame Ethernet. La présence d'un code correcteur CRC n'est pas toujours suffisant en cas de collision. Un mécanisme de détection de collision (CSMA/CD) a donc été ajouté au standard [Mol85]. En effet, Ethernet fonctionne en meilleur effort. Les trames ne sont pas assurées d'arriver à destination et n'ont pas de délai de livraison garantie. Au fil des années, IEEE 802.3 est devenue le standard prépondérant dans l'univers des réseaux câblés. Le fonctionnement d'Ethernet ne permet pas de satisfaire les contraintes des réseaux industriels. Ethernet TSN (*Time-Sensitive Networking*) a donc été créé [IL11 ; CIS17]. Il s'appuie sur les améliorations proposées à travers les sous-standards IEEE 802.1Q et 802.1AS. Ces deux sous-standards apportent entre autre la gestion du temps réel.

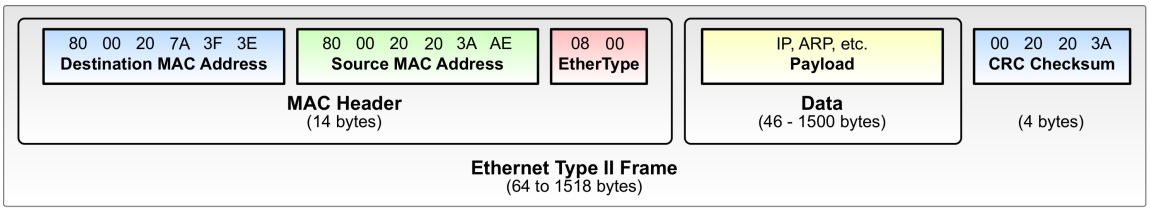


FIG. 2.1 : Structure d'une trame Ethernet extrait de *[Con18a]*

EtherCAT

EtherCAT est un protocole de niveau liaison, standardisé IEC 61158. Il a été développé par Beckhoff Automation en 2003. Il est actuellement supporté par EtherCAT Technology Group. Il s'appuie sur Ethernet pour la couche physique et la partie inférieure de la couche liaison. EtherCAT supporte les topologies de type étoile, bus et anneau [KDI10]. Il fonctionne en maître-esclaves. Le nœud nécessite seulement un système avec support temps réel et un port Ethernet. EtherCAT utilise des trames Ethernet conformes, ce qui permet en théorie d'utiliser n'importe quels câbles et cartes

réseaux compatibles avec Ethernet. En pratique, des puces FPGA et ASIC sont utilisées pour accélérer le traitement des trames sur les nœuds esclaves [Pry08]. EtherCAT utilise des paquets appelés télégrammes. Le télégramme est constitué de l'entête EtherCAT et des données (datagrammes). Il est ensuite inséré dans les trames Ethernet comme charge utile. Il est aussi possible d'encapsuler les télégrammes dans des paquets UDP [Fel05]. La Figure 2.2 illustre la structure d'un télégramme dans une trame Ethernet.

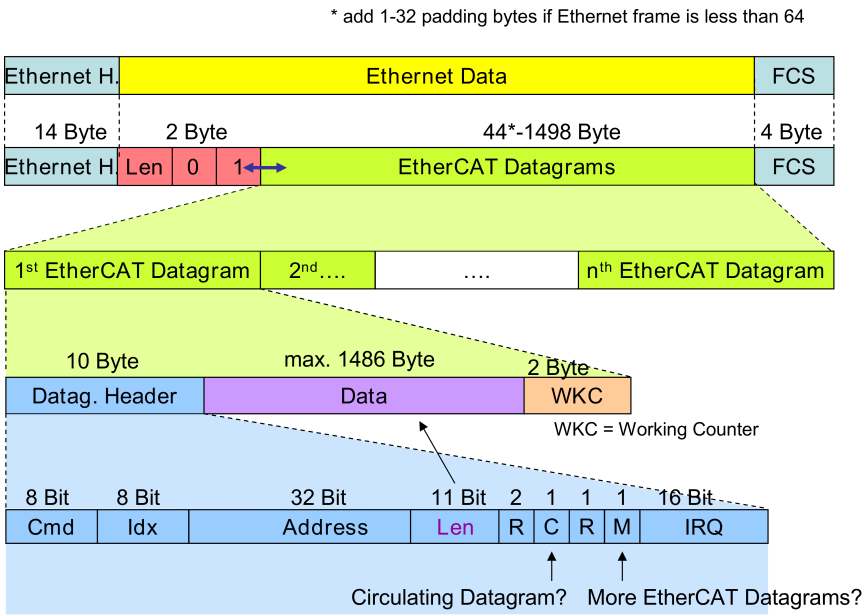


FIG. 2.2 : Structure d'une trame Ethernet contenant un télégramme EtherCAT, extrait d'ethercat.org

Concernant les performances, EtherCAT utilise au mieux *Fast Ethernet*. D'après l'article [JB04], la technologie EtherCAT serait indépendante du medium de transmission. Cette idée d'indépendance du médium est renforcée par Wu et Xie [WX17]. Ils proposent une extension d'EtherCAT pour les réseaux sans-fil IEEE 802.11 et IEEE 802.15.4. La majorité des comparatifs de performances sont purement théoriques [Pry08 ; KDI10]. Concrètement, elle utilise uniquement les spécifications pour évaluer les performances. Il n'existe pas à notre connaissance d'étude comparative indépendente sur EtherCAT. Les seules études ont été réalisées par EtherCAT Technology Group. Leur impartialité reste à démontrer.

Sercos

Sercos est un protocole de niveau liaison crée par VDW/ZVEI Joint Committee en 1987, standardisé IEC61784/61158/61800. Il est actuellement maintenu par Sercos International, basé en Allemagne. Sercos fonctionne en mode temps réel isochrone. Les équipements peuvent communiquer en fonction de cycles d'horloge. L'isochronisme implique que les cycles ont une durée strictement

identique. La dernière version est la version 3, sortie en 2003 [Sch04]. Les deux premières versions définissaient aussi la couche physique. La version 3 repose sur Ethernet pour la couche physique et la partie inférieure de la couche liaison. Sercos III supporte les topologies de type bus et anneau. Il fonctionne en mode maître-esclaves. Sercos III utilise des trames Ethernet conformes, ce qui permet en théorie d'utiliser n'importe quels câbles et cartes réseaux compatibles avec Ethernet. Des puces FPGA sont recommandées pour accélérer le traitement des trames sur les nœuds esclaves. Sercos III utilise des paquets appelés télégrammes. Le télégramme est constitué de l'entête Sercos et des données. Il est ensuite inséré dans les trames Ethernet comme charge utile. La Figure 2.3 illustre l'intégration d'un télégramme dans une trame Ethernet. Concernant les performances, Sercos III utilise uniquement *Fast Ethernet*. Nous n'avons pas trouvé d'étude permettant de comparer les performances de Sercos avec d'autres protocoles.

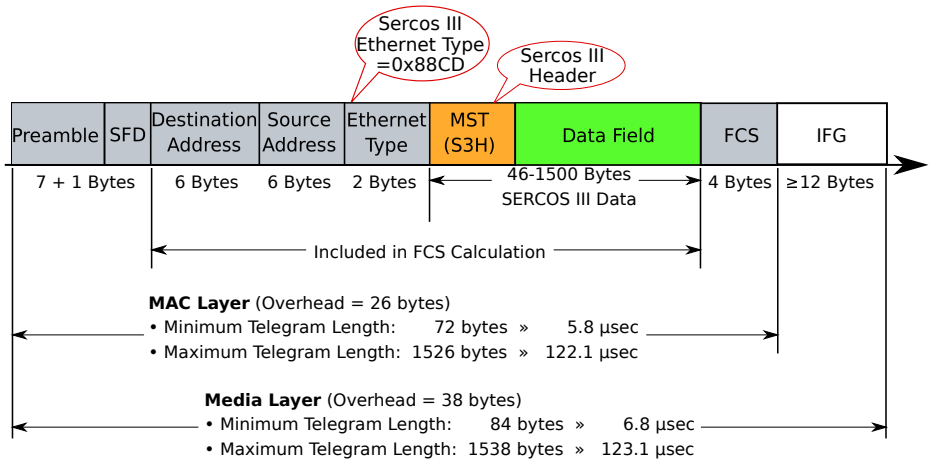


FIG. 2.3 : Structure d'une trame Ethernet contenant un télégramme Sercos III extrait de *[Con18b]*

EtherNet/IP

EtherNet/IP est un protocole de niveau applicatif (couches 5 à 7), créé par Rockwell Automation en 2001 [Bro01]. EtherNet/IP désigne l'implémentation de CIP (*Common Industrial Protocol*) sur Ethernet et IP. En 2008, des tests ont été effectués pour mettre en œuvre EtherNet/IP sur un réseau hybride Ethernet et IEEE 802.11 [CVV08]. CIP utilise TCP et UDP en fonction du besoin. EtherNet/IP et CIP sont maintenus par ODVA, Inc. (*Open DeviceNet Vendors Association*). ODVA est une organisation responsable des marques et des spécifications de CIP, EtherNet/IP, DeviceNet, ControlNet et CompoNet. ODVA est basée aux États-Unis. Peu d'informations existent sur EtherNet/IP. Bien que les spécifications soient disponibles sur le site de l'ODVA, les articles trouvés ([Sch01] et [Bro01]) sont anciens et rédigés par des employés Rockwell. À notre connaissance, il n'existe pas de comparatifs de performance.

2.2.3 Conclusions

Nous avons vu quelques technologies des réseaux câblés pouvant être présentes dans l'industrie. Nous les avons regroupées par premier niveau de fonctionnalité définie par la norme ou le standard. Certains comme RS-232 sont présentés à titre historique bien que toujours utilisés. Avant les années 2000, chaque technologie filaire propriétaire définissait sa propre couche physique. De plus, elle utilisait surtout des bus de terrain. À partir des années 2000, les nouvelles technologies câblées pour les réseaux industriels ont adopté Ethernet comme élément de base. En effet, que ce soit Sercos, POWERLINK, PROFINET ou EtherCAT, ils ont tous choisi d'utiliser en partie Ethernet comme le montre le tableau 2.1. Le support du temps réel pour Ethernet témoigne du rapprochement entre les technologies informatiques (*IT*) et les technologies opérationnelles (*OT*).

Norme	Physique	Liaison	Réseau	Transport	Application
Communication série	RS-232/RS-485	indépendant	indépendant	indépendant	indépendant
Ethernet	Ethernet	Ethernet	indépendant	indépendant	indépendant
EtherCAT	Ethernet	EtherCAT	indépendant	indépendant	indépendant
Sercos III	Ethernet	Sercos III	indépendant	indépendant	indépendant
EtherNet/IP	Ethernet	Ethernet	IP	UDP/TCP	CIP
PROFINET	Ethernet/Wifi	Ethernet/Wifi	IP	TCP/UDP	indépendant
PROFINET (NRT)	Ethernet	Ethernet	IP	UDP	RPC+PROFINET
PROFINET (RT)	Ethernet	Ethernet			PROFINET
PROFINET (IRT)	Ethernet	Ethernet modifié			PROFINET

TAB. 2.1 : Tableau récapitulatif des technologies réseaux câblés.

2.3 Wireless Personal Area Network (WPAN)

Nous commencerons par les réseaux sans-fil à faible portée (inférieur à 100 m). Ces réseaux ont des débits relativement faibles. Pour les technologies sans-fil, seules les bandes de fréquences utilisables en Europe sont indiquées, la bande de fréquences ISM 915 MHz n'étant disponible qu'aux États-Unis. Nous aborderons surtout IEEE 802.15.4 et ses dérivés : WIA-PA, ISA100.11a et WirelessHART. Nous allons commencer par détailler IEEE 802.15.4 dans la prochaine sous-section.

2.3.1 IEEE 802.15.4

IEEE 802.15.4 est un protocole de communication sans-fil standardisé par IEEE en 2003 [IEE03]. IEEE 802.15.4g [IEE+12] et IEEE 802.15.4k [IEE+13] ont une portée de l'ordre du kilomètre. Ces deux amendements s'apparentent aux technologies à faible consommation et à grande portée (*Low Power Wide Area Networks*, LPWAN) et sont donc distincts des technologies WPAN. La pile technologie ZigBee est antérieur à IEEE 802.15.4 puisque qu'elle a été développée à partir 1998 [Bak05]. IEEE 802.15.4 a été créé à partir des couches physique et liaison de ZigBee. Le standard ne définit d'ailleurs que ces deux couches. IEEE 802.15.4 fonctionne sur les bandes de fréquences ISM

868 MHz et 2,4 GHz. La portée maximale est de 100 m. Le débit maximal est de 250 kbit/s. Le débit maximal dépend de la technique d'étalement du spectre et de la bande de fréquences utilisée. IEEE 802.15.4 dispose de deux systèmes d'adressage différents. Le premier système est statique. Il utilise des adresses type EUI64, donc sur 64 bits. Le second système d'adressage est dynamique. Les adresses de ce second système sont sur 32 bits.

IEEE 802.15.4 définit plusieurs modes d'accès. Le premier est opportuniste avec évitement de collision (CSMA/CA). Alternativement, il peut aussi utiliser des balises de synchronisation (*beacon*). Il dispose aussi d'un créneau de communication garantie [KAT06]. Dans ce cas, ce mode d'accès s'apparente au fonctionnement TDMA (*Time Division Multiple Access*). En 2012, l'amendement IEEE 802.15.4e apporte des améliorations à la couche liaison [DAS14]. Il apporte notamment le mode TSCH (*Time Slotted Channel Hopping*). Il s'agit d'un fonctionnement proche du TDMA couplé à du multi-canaux et du saut de fréquences. Avec ce mode de fonctionnement, IEEE 802.15.4e s'inspire des normes pour les réseaux industriels WIA-PA, ISA100.11a et WirelessHART que nous allons présenter dans les sous-sections suivantes. Les ajouts d'IEEE 802.15.4e en fait un standard compatible avec les attentes d'un réseau industriel.

Il est possible d'utiliser IPv6 sur IEEE 802.15.4. Néanmoins, il est nécessaire d'apporter une couche d'adaptation. Cette couche d'adaptation s'appelle 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*). 6LoWPAN est un standard IETF issu de la RFC 4944 [Mon+07].

2.3.2 WIA-PA

WIA-PA (*Wireless Networks for Industrial Automation–Process Automation*) est un protocole réseau sans-fil. WIA-PA a été développé et approuvé en 2007 par la CIWA (*Chinese Industrial Wireless Alliance*) [Lia+11]. Il faut attendre 2011 pour que l'IEC le standardise sous la norme IEC 62601. WIA-PA s'appuie sur la couche physique et la partie inférieure (MAC) du standard IEEE 802.15.4. Concrètement, il reprend la couche physique, le système d'adressage et la structure des trames. WIA-PA étend certaines fonctionnalités d'IEEE 802.15.4, notamment les super-trames. Il ajoute aussi un mécanisme de saut de fréquences [Lia+13]. WIA-PA définit aussi la couche réseau et application. L'adresse de niveau réseau n'est constituée que de 16 bits. Le format de l'adresse est de type X.Y où X et Y sont compris entre 0 et 255. La couche réseau supporte l'agrégation des paquets et des données.

2.3.3 ISA100.11a

ISA100.11a est une pile technologique réseau sans-fil développée par l'ISA (*International Society of Automation*) en 2009. En 2010, il a été standardisé par l'IEC sous le nom d'IEC 62734. ISA100.11a s'appuie sur la couche physique et la partie inférieure de la couche liaison d'IEEE 802.15.4 [PC11]. ISA100.11a définit le reste de la couche liaison, la couche réseau, transport et application. Comme WIA-PA, ISA100.11a ajoute le saut de fréquences [Car08]. Il est possible d'utiliser TDMA pour maximiser l'utilisation du média. La couche réseau est assurée par IPv6 et sa couche d'adaptation 6LoWPAN. Cela permet de rendre le routage possible vers l'extérieur du réseau ISA100.11a. Néanmoins,

pour des raisons d'efficience, les adresses IP sont sur 16 bits à l'intérieur du réseau ISA100.11a. Le format d'adresse IPv6 standard reste disponible si besoin. La couche transport utilise UDP avec des extensions [PC11 ; NR12]. La couche applicative utilise EDDL (*Electronic Device Description Language*, IEC 61804). Ce langage de description est utilisé par d'autres technologies comme HART et Foundation Fieldbus [Car08].

2.3.4 WirelessHART

WirelessHART est une pile technologique réseau sans-fil développée par HART Communications Foundation et publiée en 2007 [Kim+08]. WirelessHART est standardisé IEC 62591 en 2010. WirelessHART reprend la couche physique et la partie inférieure (MAC) de la couche liaison d'IEEE 802.15.4. Il définit le reste de la couche liaison, la couche réseau, transport et application. Afin d'avoir un comportement plus déterministe, WirelessHART utilise uniquement le fonctionnement TDMA à la place de CSMA [CNM10]. Il ajoute aussi le support pour le saut de fréquences. La couche transport utilise UDP et TCP en fonction du besoin. Pour la couche application, WirelessHART reprend simplement HART.

2.3.5 Conclusions

Nous avons différentes technologies réseaux WPAN. La portée maximale est d'environ 100 m. Leur débit ne dépasse pas les 2 Mbit/s. Leur objectif est avant tout de maximiser la durée de vie des équipements connectés. ISA100.11a, WIA-PA et WirelessHART reprennent partiellement le standard IEEE 802.15.4. Au niveau de la couche physique, leur portée et leur débit sont identiques. Ce sont les couches supérieures qui les distinguent les uns des autres. Le tableau 2.2 récapitule brièvement les informations techniques des technologies rencontrées.

Technologie	Bandes de fréquences	Méthode de contrôle d'accès
IEEE 802.15.4	868 MHz / 2,4 GHz	TDMA ou CSMA
WIA-PA	2,4 GHz	TDMA+FDMA ou CSMA
ISA100.11a	2,4 GHz	TDMA ou CSMA
WirelessHART	2,4 GHz	TDMA ou CSMA

Tab. 2.2 : Tableau récapitulatif des technologies WPAN.

2.4 Wireless Local Area Network

Nous avons vu des réseaux à faible portée. Leur portée n'excède pas 100 m. Nous allons nous intéresser aux technologies réseaux intermédiaires. Dans cette catégorie, il n'existe pour le moment qu'une seule famille : les normes IEEE 802.11. En effet, les normes HiperLAN proposées par ETSI (*European Telecommunications Standards Institute*) n'ont pas rencontrées le succès espéré et ont été abandonnées.

2.4.1 IEEE 802.11

IEEE 802.11 a été standardisé par l'IEEE en 1997. Wi-Fi définit à la fois la couche physique et la couche liaison [DW93]. Bien que la norme dépende de l'IEEE, le nom Wi-Fi (ou wifi) est une marque déposée en 1999. Il est détenu par Wi-Fi Alliance. La norme initiale (IEEE 802.11-1997) fonctionnait sur la bande de fréquences ISM 2,4 GHz [IL97]. Le débit était de 2 Mbit/s. En pratique, cette norme a été très peu utilisée. La deuxième version d'IEEE 802.11 (IEEE 802.11a) fonctionne sur la bande de fréquences ISM 5 GHz [IEE+99]. Son débit maximum théorique dans sa dernière révision est de 54 Mbit/s, ce qui la rend nettement plus intéressante que la version précédente. Le tableau 2.3 récapitule les informations essentielles sur les différentes normes IEEE 802.11.

Normes	Nom	Année	Bande de fréquences	Débit théorique max.	Portée max. (m)
IEEE 802.11-1997		1997	2.4 GHz	2 Mb/s	100
IEEE 802.11a		1999	5 GHz	54 Mb/s	120
IEEE 802.11b		1999	2,4 GHz	11 Mb/s	140
IEEE 802.11g		2003	2,4 GHz	54 Mb/s	140
IEEE 802.11n	Wi-Fi 4	2009	2,4 GHz/5 GHz	600 Mb/s	250
IEEE 802.11ad	WiGig	2012	60 GHz	6,7 Gb/s	10
IEEE 802.11ac	Wi-Fi 5	2013	5 GHz	3,4 Gb/s	300
IEEE 802.11af		2013	470 MHz-710 MHz	12 Mb/s	qq km
IEEE 802.11ah	Wi-Fi HaLow	2016	868 MHz	8 Mb/s	1000
IEEE 802.11ax	Wi-Fi 6/6E	2021	2,4 GHz/5 GHz/6 GHz	10,5 Gb/s	300

TAB. 2.3 : Tableau récapitulatif des normes 802.11

Au niveau de la couche liaison, IEEE 802.11 est proche d'Ethernet. Les données transitent sous forme de trames. Les terminaux possèdent une adresse sur 48 bits, dépendante du constructeur. IEEE 802.11 supporte différentes topologies. La topologie la plus commune est celle en étoile. Le point d'accès est au centre de l'étoile. IEEE 802.11 peut aussi fonctionner au mode ad-hoc. Xu *et al.* [XS01] estiment que la couche liaison n'est pas pour autant optimale. IEEE 802.11 fonctionne en mode CSMA/CA puis une variante à partir d'IEEE 802.11e [IEE05]. En 2017, Cheng *et al.* [CYZ17] proposent d'utiliser le fonctionnement TDMA au lieu du CSMA/CA. Ce changement rend IEEE 802.11 déterministe à condition de synchroniser les éléments du réseau. D'après les essais de Cheng *et al.*, le fonctionnement TDMA permet une amélioration des performances en multi-sauts car il diminue le nombre de retransmissions. IEEE 802.11ax utilise OFDMA au lieu d'une version améliorée de CSMA/CA [Bel16].

Wifi et la sécurité

Au cours de son évolution, différentes sécurités ont été intégrées à IEEE 802.11. La première d'entre elle est WEP (*Wired Equivalent Privacy*). WEP est disponible depuis la norme initiale de 1997.

La sécurité WEP utilise un système de clé partagée pour l'authentification. La taille de la clé n'est que de 64 bits (clé de 40 bits + vecteur d'initialisation (IV) de 24 bits), ce qui est très insuffisant de nos jours. Elle utilise RC4 pour le chiffrement du flux de données. L'intégrité est réalisée par contrôle CRC-32. En 2001, Fluhrer, Mantin et Shamir [FMS01] proposent une première cryptanalyse de WEP. Cette attaque consiste en une première phase d'écoute, puis une phase d'attaque sur les trames écoutées. Dans de bonnes conditions, ces deux phases permettent de récupérer la clé partagée en quelques minutes. D'autres faiblesses ont été découvertes par la suite. L'augmentation de la taille de la clé à 128 bits (clé de 104 bits + IV de 24 bits) n'a pas permis de combler ces faiblesses.

Suite à la publication des premières cryptanalyses de WEP, le protocole de sécurité WPA (*Wi-Fi Protected Access*) fut publié en 2003. WPA adopte le système d'authentification par clé pré-partagée et EAP (*Extensible Authentication Protocol*). WPA intègre aussi TKIP (*Temporal Key Integrity Protocol*). RC4 est toujours utilisé, par contre la taille de clé est de 128 bits et la taille de l'IV de 48 bits. CRC-32 est remplacé par Michael [Fer02]. Malheureusement, comme TKIP reprend la conception de WEP, une première cryptanalyse est publiée en 2008 [TB09]. En 2004, WPA2 remplace WPA. TKIP est remplacé par CCMP, un protocole s'appuyant sur AES. Pour autant, WPA2 n'est pas infaillible [VP17]. Wi-Fi Alliance a annoncé WPA3 en 2018. Malheureusement, un an plus tard, la vulnérabilité *Dragonblood* a été découverte dans WPA3 [VR19]. Celle-ci permet notamment d'obtenir la clé d'authentification via une attaque par canal auxiliaire.

802.11af et 802.11ah

Les normes 802.11af et 802.11ah sont des normes un peu particulières. Elles ont des débits maximums plus faibles que les autres normes et une plus grande portée. De plus, elles opèrent à des fréquences inférieures à 1 GHz. À l'exception d'IEEE 802.11af [Flo+13], Wi-Fi fonctionne uniquement sur les bandes de fréquences non-licenciées ISM (Industrielle, Scientifique et Médicale). Elle fonctionne entre 470 MHz et 710 MHz, c'est-à-dire une bande de fréquences réservée à la diffusion télévisuelle. IEEE 802.11af utilise donc la radio cognitive pour récupérer des bandes de fréquences inutilisées. Le débit est donc dépendant de l'utilisation du spectre. Néanmoins, il ne faut pas espérer des débits supérieurs à 12 Mbit/s [LM12]. Sa portée peut atteindre quelques kilomètres.

IEEE 802.11ah est la norme destinée à l'IoT et aux M2M (machine à machine) [Ada+14 ; Kho+15]. Elle porte le nom commercial de Wi-Fi HaLow. Le débit est volontairement plus faible afin d'augmenter la durée de vie des équipements sur batteries. Sa portée permet d'avoir une solution intermédiaire avant les réseaux cellulaires. IEEE 802.11ah utilise la bande de fréquences ISM 868 MHz en Europe.

2.5 Conclusions

Nous avons vu plusieurs technologies réseaux filaires et sans-fil. Les technologies filaires s'articulaient autour de solutions propriétaires et notamment des bus de terrain. Depuis les années 2000, les nouvelles technologies filaires s'appuient sur Ethernet. Elles peuvent néanmoins garder des éléments propriétaires. Mais il s'agit du premier rapprochement entre les réseaux industriels et

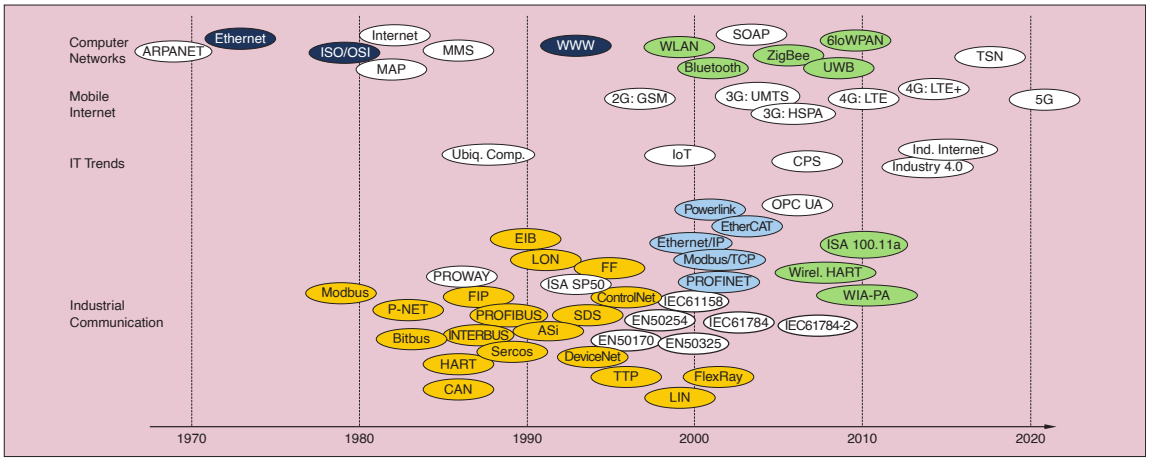


FIG. 2.4 : Le rapprochement entre les technologies *IT* et *OT*, extrait de [WSJ17]

les réseaux informatiques. Avec les amendements IEEE 802.1Q et 802.1AS, Ethernet se rapproche des technologies des réseaux industriels. En effet, ces deux ensembles d'amendements apportent le support du temps réel nécessaire aux applications industrielles.

Une situation similaire s'est produit avec les technologies sans-fil. Des technologies sans-fil propriétaires se sont s'appuyées sur IEEE 802.15.4 mais en apportant les éléments nécessaires aux applications industrielles. Quelques années plus tard, IEEE 802.15.4 a intégré le saut de fréquences et le découpage temporel, des fonctionnalités initialement absentes mais présentes dans les technologies sans-fil propriétaires. La Figure 2.4 illustre ce rapprochement entre les technologies issues des réseaux informatiques (*IT*) et celles des réseaux industriels (*OT*), que ce soit pour les technologies filaires ou sans-fil. Les technologies des réseaux informatiques ne sont pas les seules à s'adapter progressivement aux contraintes des réseaux industriels. Les réseaux cellulaires privés de 5G offriront des solutions pour l'industrie du futur grâce à leur large couverture et leur faible latence [Aij20].

Malgré toutes ces avancées, les performances des réseaux sans-fil peuvent être dégradées par les interférences. Elles peuvent être causées par l'environnement et par la concurrence inter-protocole. En effet, parmi les technologies sans-fil abordées, certaines comme IEEE 802.11 et IEEE 802.15.4 utilisent la même bande de fréquences. Dans le prochain chapitre, nous verrons plus en détails les interférences et des solutions pour optimiser les réseaux notamment sans-fil.

Chapitre 3

Vers des réseaux intelligents

Sommaire

4.1 Introduction **53**

4.2 Apprentissage par renforcement **53**

4.2.1 Quelques exemples d'algorithmes d'apprentissage par renforcement . . 55

4.3 Applications de l'apprentissage par renforcement aux problèmes de réseaux informatiques **58**

4.3.1 Application au paramétrage automatique du protocole TCP 58

4.3.2 Application à la fonction de routage 59

4.4 Q-routing **60**

4.4.1 Fonctionnement de Q-routing 60

4.4.2 Analyse des résultats obtenus par Boyan et Littman 61

4.5 Modifications génériques de Q-routing **63**

4.5.1 Predictive Q-routing 63

4.5.2 Dual Reinforcement Q-Routing 63

4.5.3 Q-Neural Routing 64

4.5.4 K-Shortest path Q-routing 64

4.5.5 Q^2-routing 64

4.6 Adaptation de Q-routing à des scénarios particuliers **65**

4.6.1 Q-routing sur des scénarios avec mobilité 65

4.6.2 Q-routing pour les réseaux de capteurs 66

4.6.3 Q-routing pour la radio cognitive 66

4.7 Conclusions **66**

3.1 Introduction

Dans ce chapitre, nous abordons le problème des interférences dans les réseaux sans-fil. L'atténuation des effets des interférences permet déjà d'améliorer les performances des réseaux sans-fil. Dans la Section 3.2, nous verrons les interférences et leurs effets. Mais l'atténuation des interférences est surtout valable pour les réseaux sans-fil. Mais, il n'est pas toujours possible de diminuer les interférences, à cause de l'environnement par exemple. Il faut donc utiliser d'autres moyens d'améliorer les performances comme la création d'une optimisation inter-couche (Section 3.3). Une optimisation inter-couche consiste à s'affranchir de la modélisation OSI afin d'améliorer les performances. Avec une optimisation inter-couche, il est donc possible de faire communiquer plusieurs couches entre elles. Enfin, les réseaux cognitifs sont une généralisation de la radio cognitive à l'ensemble des couches de la modélisation OSI. Ils utilisent potentiellement des optimisations inter-couches afin d'atteindre un ou plusieurs objectifs de performances. Les réseaux cognitifs occuperont la Section 3.4.

3.2 Les problèmes d'interférences radio

Nous allons commencer par voir le phénomène d'interférence radio. Il existe plusieurs sources d'interférence. Nous allons en voir deux types : *i*) les interférences par multi-trajets et *ii*) les interférences multi-utilisateurs. Les conditions de création sont différentes. Les interférences par multi-trajets sont créées à cause de l'environnement. Dans le cas des interférences multi-utilisateurs, les autres utilisateurs de la bande de fréquences en sont la source. Nous verrons ensuite plusieurs solutions pour atténuer ce phénomène d'interférences comme les récepteurs RAKE ou le routage optimisé. Enfin, nous terminerons avec les effets des interférences et notamment la concurrence inter-protocole.

3.2.1 Interférences multi-trajets

Le premier cas est celui des interférences par multi-trajets (*multipath*) [SV87]. Ces interférences se créent lorsqu'une même onde prend plusieurs chemins. Lorsque ces trajets sont de longueur différente, cette même onde radio arrive plusieurs fois au destinataire. Si l'espace temporel entre chaque train d'onde est suffisamment important, il n'y a pas d'incidence car même les ondes passant par le chemin le plus long seront arrivés avant le train d'ondes suivant. Si ce n'est pas le cas, un train d'ondes précédemment reçu peut être de nouveau reçu en même temps que le train d'ondes suivant. Il y a création d'une interférence multi-trajets. La formation d'interférences multi-trajets dépend surtout de l'environnement. Car dans un environnement sans surface réfléchissante, les ondes ne prennent qu'un seul chemin.

La solution la plus simple pour lutter contre les interférences multi-trajets est d'utiliser un récepteur RAKE [BOW00]. Un récepteur RAKE est composé de plusieurs sous-récepteurs. Le signal est reconstruit à partir de l'ensemble des signaux reçus par les sous-récepteurs. Indépendamment du matériel utilisé, la méthode d'étalement du spectre peut avoir une incidence sur les interfé-

rences multi-trajets. En 1994, Tsai et Chang [TC94] proposent d'utiliser un étalement de spectre de type CSS (*Chirp Spread Spectrum*). Cet étalement est notamment utilisé aujourd'hui par le protocole sans-fil LoRa [Sem20]. MC-CDMA (*Multi-Carrier Code-Division Multiple Access*) est évoqué comme étant une autre solution potentielle dans [YL94]. MC-CDMA découpe la porteuse principale en sous-porteuses orthogonales. Ces sous-porteuses sont moins sensibles aux interférences multi-trajets à cause de leur orthogonalité. Il existe un autre type d'interférence : les interférences multi-utilisateurs.

3.2.2 Interférences multi-utilisateurs

Nous allons maintenant aborder un autre type d'interférence. La présence de plusieurs utilisateurs sur un réseau peut créer des interférences. Ce cas peut se produire lorsque deux terminaux essaient de communiquer en même temps, car chacun peut créer une onde de fréquence similaire. Il s'agit des interférences multi-utilisateurs (IMU). D'après [Nob03], l'utilisation de codes orthogonaux suffit dans la plupart des cas à résoudre le problème. Néanmoins, en cas de trajets multiples, cette solution seule devient insuffisante. En effet, l'orthogonalité peut être brisée par les réflexions des ondes, des IMU peuvent réapparaitre. Les méthodes d'accès multiple par répartition dans le temps permettent de résoudre le problème des IMU. Par exemple, dans un réseau en mode TDMA, chaque terminal ne peut émettre qu'à des moments bien précis. Ainsi, il n'est normalement pas possible que deux terminaux émettent en même temps. Malheureusement, le fonctionnement TDMA est très peu efficient en termes d'utilisation du média.

3.2.3 Gestion des collisions

En réseaux informatiques, les collisions sont une conséquence du phénomène d'interférences. Nous avons vu que le fonctionnement en mode TDMA pouvait résoudre normalement le problème des interférences multi-utilisateurs. Néanmoins, le fonctionnement en mode TDMA est peu efficient dans certaines situations. CSMA (*Carrier Sense Multiple Access*) est un système de fonctionnement plus opportuniste donc plus propice aux collisions. Dans les réseaux câblés, CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*) permet de détecter une collision et d'éviter qu'elle se reproduise grâce au mécanisme d'émission différée [TB80]. Pour les réseaux sans-fil, il n'est pas toujours possible de détecter une collision notamment dans le cas de terminal caché [Kar90]. Par contre, il est possible d'éviter les collisions. Cette variante de CSMA s'appelle CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*). Dans IEEE 802.11, deux trames (*Request To Send*, RTS et *Clear To Send*, CTS) sont utilisées pour éviter les collisions [CBV04 ; JA07]. Le terminal demande d'abord l'autorisation de communiquer (trame RTS). Un terminal maître, généralement le point d'accès, autorise ou non le terminal à communiquer (trame CTS).

3.2.4 Protocole de routage optimisé

Nous avons vu que les méthodes de contrôle d'accès au support avaient une importance dans la gestion des interférences. Les interférences multi-utilisateurs touchent particulièrement les ré-

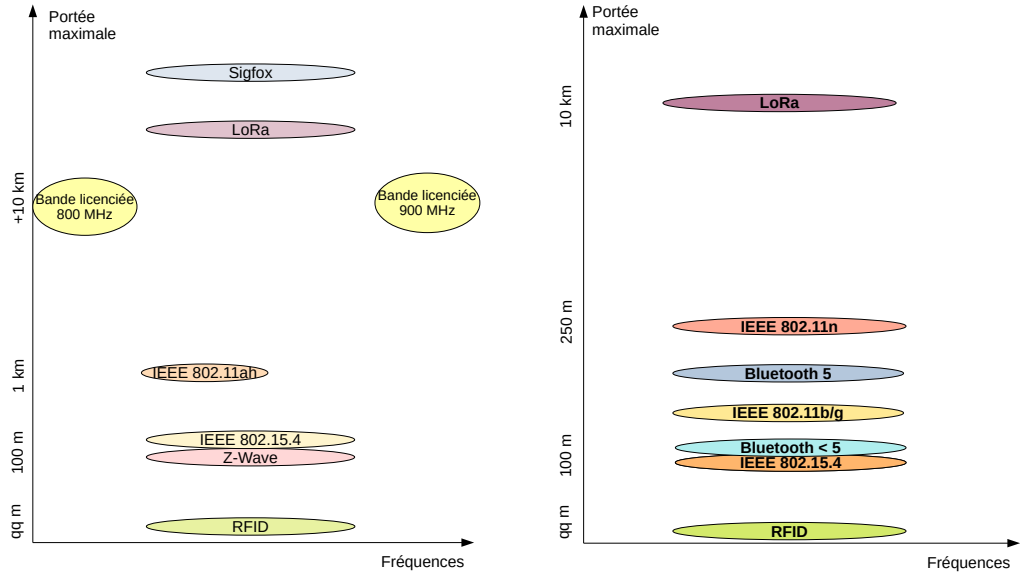
seaux ad-hoc et de type maillage (*mesh*). Quelques travaux [Ric+05 ; SBM06 ; Bru07] ont montré que le routage peut avoir une incidence sur la création d'interférences. Le chemin optimal selon un algorithme de routage classique n'est pas nécessairement celui qui crée le moins d'interférences. À cet effet, l'article [SBM06] et le brevet [BMS13] proposent des protocoles de routage optimisé en fonction des interférences en réseaux maillés.

Dans une certaine mesure, OLSR (*Optimized Link State Routing Protocol*, RFC 3626) peut orienter son routage de façon à minimiser les interférences. La métrique ETX (*Expected Transmission Count*) évalue la qualité du lien entre deux nœuds [MN09]. En effet, cette métrique est le ratio entre le nombre de paquets et le nombre de transmissions nécessaires pour les envoyer. Si cette métrique est prépondérante par rapport au nombre de sauts, OLSR favorisera les liens créant le moins d'interférences car ils minimisent le nombre de retransmissions. L'article [SBM06] montre avec AODV (*Ad-hoc On-demand Distance Vector*, RFC 3561) que les métriques ETT (*Expected Transmission Time*, métrique ETX pondérée par le ratio taille d'envoi sur bande passante) et WCETT (*Weighted Cumulative Expected Transmission Time*) permettent d'obtenir des meilleurs débits par rapport au nombre de sauts en réseau maillé. Cela illustre que le protocole de routage et surtout la métrique de routage permettent d'obtenir de meilleures performances sans pour autant changer le matériel ou la technologie utilisée.

Les protocoles de routage multi-chemins sont parfois accusés de créer des interférences car ils peuvent utiliser des chemins rentrant en concurrence. Des protocoles de routages multi-chemins minimisant les interférences ont alors été conçus [LC04 ; THT08 ; LPN11]. Par exemple, Le *et al.* [LPN11] proposent IA-MPOLSR (*Interference Aware Multi-Path Optimized Link State Routing Protocol*), une version de MP-OLSR (*Multi-Path Optimized Link State Routing Protocol*, RFC 8218) minimisant les interférences. L'étude propose des comparaisons avec OLSR mais pas avec la version originale de MP-OLSR [Yi+11]. Il est difficile de montrer que les protocoles de routages multi-chemins provoquent une augmentation substantielle des interférences. Néanmoins, selon la topologie, les protocoles de routages multi-chemins peuvent créer des conditions favorables aux interférences multi-utilisateurs.

3.2.5 Concurrence inter-protocoles

La présence d'interférences est due à l'utilisation de bandes de fréquences proches. De nombreux protocoles fonctionnent sur les bandes de fréquences ISM 868 MHz et 2,4 GHz. Ils peuvent donc interférer entre eux. La Figure 3.1 représente les différentes technologies réseaux par bandes de fréquences ISM en fonction de leur portée maximale. Concernant les fréquences de fonctionnement, la Figure 3.1 ne suit pas rigoureusement la répartition des fréquences. Par exemple, RFID fonctionne que sur une petite partie de la bande ISM 2,4 GHz (entre 2446 et 2454 MHz). Le décret ARTL1430373S [Leg15] précise les modalités exactes d'utilisation des bandes ISM en termes de bandes de fréquences, puissance d'émission et coefficient d'utilisation limite (*duty cycle*) en fonction de leur catégorisation.



(a) Sur la bande ISM 868 MHz (b) Sur la bande ISM 2,4 GHz

FIG. 3.1 : Représentations des différentes technologies réseaux en fonction de la fréquence et de leur portée.

Bande ISM 868 MHz

La bande ISM 868 MHz est propice aux interférences. En effet, elle est déjà utilisée pour de nombreux équipements. Ces équipements sont rangés en trois catégories : *i)* RFID,*ii)* les dispositifs à faible coefficient d'utilisation/à haute fiabilité et *iii)* les dispositifs à courte portée non spécifiques. Chaque catégorie possède des conditions d'utilisation du spectre qui lui est propre. La Figure 3.2 montre la fragmentation de la bande ISM 868 MHz, les différents coefficients d'utilisation et la puissance apparente rayonnée maximale pour les dispositifs à courte portée non spécifiques. Elle n'inclut donc pas la RFID, ni les dispositifs à faible coefficient d'utilisation/à haute fiabilité (par exemple les alarmes). De plus, des interférences peuvent apparaître à cause des bandes de fréquences licenciées 800 MHz et 900 MHz. Les parties adjacentes des bandes licenciées à la bande ISM 868 MHz sont utilisées uniquement en *uplink*. Concrètement, il n'y a que les terminaux mobiles qui puissent interférer sur la bande ISM 868 MHz. En 2017, Lauridsen *et al.* [Lau+17] ont conduit une étude sur les risques d'interférences en condition réelle sur LoRa et Sigfox. Ils ont réalisé leurs mesures dans trois zones aux caractéristiques distinctes : un centre-ville, une zone résidentielle et une zone industrielle. D'après leurs mesures, la probabilité d'interférences se situe en moyenne entre 22 et 33 %. Dans la zone industrielle, le risque d'interférences peut être lié aux systèmes RFID d'après l'étude. D'après la décision 2006/771/CE modifiée, les appareils RFID peuvent utiliser une puissance d'émission maximale de 2 W sur des canaux bien précis de la bande ISM 868 MHz. Ce qui peut contribuer à expliquer le risque d'interférence. Vejlgaard *et al.* [Vej+17] ont montré que les interférences peuvent

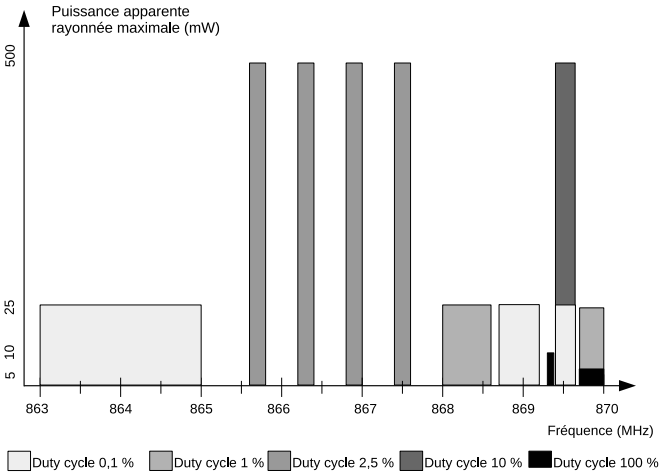


FIG. 3.2 : Portion de la bande ISM 868 MHz utilisable pour des applications réseaux.

réduire la portée de 10 % en extérieur et jusqu'à 50 % en intérieur des solutions LoRa et Sigfox. Le taux de paquet perdu ou invalide peut atteindre 60 % pour Sigfox et 50 % pour LoRa que ce soit pour le trafic montant ou descendant.

Bande ISM 2,4 GHz

La bande ISM 2,4 GHz est aussi propice aux interférences. Contrairement à la bande ISM 868 MHz, la bande ISM 2,4 GHz est disponible partout dans le monde. De nombreux protocoles utilisent cette bande de fréquences. Sikora et Groza [SG05] ont effectué des tests sur les interférences entre des équipements IEEE 802.15.4 (à 2,4 GHz, étalement de type DSSS, *Direct Sequence Spread Spectrum*) et d'autres équipements pouvant interférer. Si IEEE.15.4 est en concurrence avec IEEE 802.11b sur la même gamme de fréquences, quasiment 90 % des trames IEEE 802.15.4 sont détruites. Le résultat est similaire si IEEE 802.15.4 entre en concurrence avec IEEE 802.11n [Pet+07]. IEEE 802.11g devrait avoir les mêmes effets car il a les mêmes caractéristiques techniques. IEEE 802.11a et ac n'ont aucun effet car ils n'utilisent pas la même bande de fréquences. Bluetooth et les fours à micro-ondes ont un impact raisonnable sur IEEE 802.15.4, environ 10 % de trames détruites. En effet, les fours à micro-ondes émettent des ondes sur la bande ISM 2,4 GHz lors de leur fonctionnement [KE97]. L'occupation de la bande ISM est dépendante du moment de la journée et du jour de la semaine [BHC04]. L'occupation est maximale les jours ouvrés pendant les heures d'activité.

Les réseaux 5G privés

Les réseaux 5G privés peuvent utiliser certaines bandes de fréquences licenciées et les bandes ISM 2,4 GHz, 5 GHz et 6 GHz [Aij20]. La bande ISM 2,4 GHz est déjà très occupée comme nous le

précisions précédemment. Les bandes ISM 5 GHz et 6 GHz sont notamment utilisées par plusieurs amendements d'IEEE 802.11 (a, n, ac et ax).

3.3 Optimisation inter-couches

Dans cette section, nous abordons l'optimisation inter-couches (*cross-layer optimisation*). L'optimisation inter-couches repose sur le concept de conception inter-couches (*cross-layer design*). Nous commencerons par le principe de l'optimisation inter-couches. Nous poursuivrons avec des exemples d'utilisation et d'implémentation. Enfin, nous terminerons avec les limites de l'optimisation inter-couches.

Avant de définir l'optimisation inter-couches, nous allons commencer par définir le contexte de sa création. Dans les réseaux informatiques, le modèle OSI (*Open Systems Interconnection*) est le modèle prépondérant [Zim80]. Ce modèle est considéré comme fondamental. Mais, le modèle OSI impose une stricte séparation entre chaque couche de service. Seule une communication limitée entre deux couches adjacentes est tolérée. Ce principe permet de garantir une certaine compatibilité et indépendance entre chaque protocole. Néanmoins, ce manque de flexibilité rend impossible les interactions inter-couches surtout quand celles-ci ne sont pas adjacentes. C'est pour cela que l'optimisation inter-couches est apparue.

3.3.1 Le principe de l'optimisation inter-couches

Une conception de protocole s'affranchissant des principes de l'architecture OSI est une conception inter-couches [SM05]. Srivastava et Motani [SM05] notent plusieurs transgressions possibles. La première consiste à créer de nouvelles méthodes de communication, par exemple à créer de nouvelles interfaces entre les couches ou partager des paramétrages à travers plusieurs couches. Il est aussi possible de créer un protocole pouvant être utilisé sur différentes couches. L'optimisation inter-couches a pris de l'importance avec l'essor des réseaux sans-fil. La principale raison est que le modèle OSI a été créé pour les réseaux filaires. La plupart des technologies sans-fil utilisent une conception inter-couches. Elles définissent généralement la couche physique et la couche liaison. Par exemple, la nature du média n'est pas prise en compte, d'autant que certaines caractéristiques physiques comme la densité varient au cours du temps.

Il existe deux grandes catégories d'optimisation inter-couches [Gon05] : *i) joint-layer design* utilisant des techniques d'optimisation et *ii)* les conceptions inter-couches adaptatives. La première catégorie repose sur la communication entre deux ou plusieurs couches et utilise une technique d'optimisation. Par exemple, ElBatt et Ephremides [EE04] proposent un framework qui combine la gestion de l'alimentation et le fonctionnement en TDMA. Leur objectif est de maximiser le nombre de communications simultanées pendant un même créneau en respectant la contrainte de garder un ratio signal sur interférences et bruit (*Signal-to-Interference-plus-Noise Ratio*, SINR) minimum prédéfini. Malgré les nombreuses hypothèses formulées (14 en tout), les auteurs ont dû diviser en deux phases leur algorithme. D'après Gong [Gon05], les solutions de cette catégorie souffrent de

leur grande complexité ainsi que de leur problème de convergence. De plus, lors de leur conception, des hypothèses simplificatrices peu réalistes sont appliquées. La seconde catégorie regroupe les conceptions inter-couches adaptatives. Par exemple, une optimisation qui utiliserait les informations de la couche physique pour paramétrer la couche liaison pour maximiser le débit.

Il existe plusieurs solutions pour créer une conception inter-couches [RI04]. La Figure 3.3 illustre les différents schémas possibles. Pour les cas de A à C, l'optimisation repose sur la création de nouvelles interfaces de communication inter-couches. Par exemple, Xylomenos et Polyzos [XP01] proposent que les applications informent la couche liaison que les informations sont sensibles aux délais. La couche liaison peut alors prioriser les trames venant de ces applications en complément de la priorisation faite sur la couche réseau. Le cas D montre la possibilité de fusionner plusieurs couches entre elles pour ne former qu'une super-couche. Le cas E consiste à créer une couche en fonction du fonctionnement d'une couche bien précise. Enfin, le cas F revient à optimiser toute la pile de communication. Ce dernier cas est normalement celui qui offre les meilleures performances. Néanmoins, cette optimisation globale s'effectue au détriment d'autres paramètres. Par exemple, il faut que toutes les couches supportent le paramétrage. L'ensemble de la pile est finalement relativement couplé. C'est le cas pour la première version de ZigBee qui ne formait qu'une seule pile.

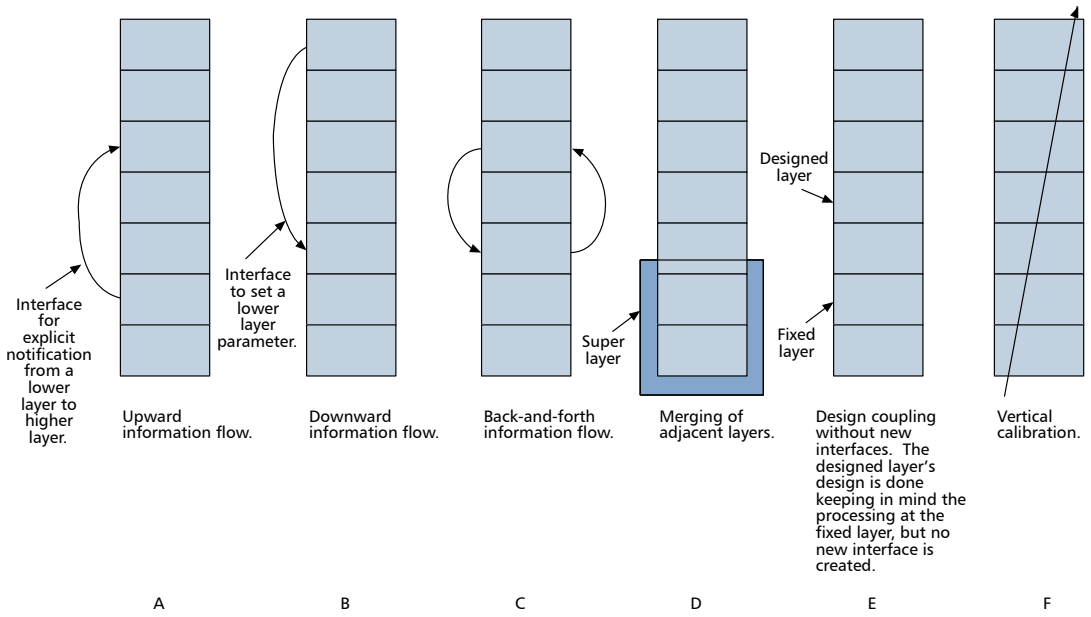


FIG. 3.3 : Les 6 différentes possibilités de conception inter-couches identifiées par A, B, C, D, E, F, extrait de [SM05].

Il existe plusieurs moyens pour faire communiquer les différentes couches entre elles. Les interfaces sont une première méthode. Une interface définit au préalable quelles sont les différentes interactions possibles avec les autres couches. Elle définit aussi les paramètres modifiables ou simplement accessibles. Une seconde méthode consiste à créer un point d'échange partagé entre les

différentes couches. Ce point d'échange permet à chaque couche de mettre à disposition les informations qu'elle souhaite. Ce point d'échange peut prendre la forme d'une base de données partagée par exemple [SM05]. La dernière méthode consiste à ajouter des éléments d'abstractions. Ces éléments d'abstractions font l'interface entre deux couches sans que les couches impliquées ne connaissent l'interface de l'autre [SM05].

3.3.2 Exemples d'application

Nous avons défini les principaux concepts de l'optimisation inter-couches. Nous allons maintenant aborder quelques cas concrets de mise en application.

Radio cognitive

La radio cognitive est un exemple d'optimisation inter-couches. En effet, à minima, les couches physique et liaison communiquent entre elles. Nous avons vu qu'il y a un fort couplage entre ces couches. En effet, la couche liaison est fortement dépendante de la couche physique que ce soit pour la gestion des bandes de fréquences à utiliser ou le balayage et la détection des utilisateurs primaires (*Primary User*, PU). Pour encore augmenter les performances, il existe aussi des protocoles de routage optimisés pour la radio cognitive[LCS12]. Il s'agit d'une optimisation inter-couches entre la couche physique et la couche réseau. Luo *et al.* [Luo+10] ont été plus loin. Leur proposition consiste à améliorer les performances de TCP sur les réseaux utilisant la radio cognitive. D'après leur simulation sous *ns-2*, leur optimisation permet un débit avec un gain d'environ 10 % par rapport l'utilisation de TCP classique.

Réseaux ad-hoc

Les réseaux ad-hoc sont des réseaux ne reposant pas sur une infrastructure. Chaque nœud participe activement au réseau. Les travaux sur les optimisations inter-couches portent surtout sur les réseaux ad-hoc mobiles (*Mobile Ad-hoc NETworks*, MANET). À partir de 2002, un projet européen nommé MobileMAN a été financé dans le cadre de l'appel à projet « Information Society Technologies ». Ce projet est piloté par le conseil national italien de la recherche et réunit notamment l'université de Cambridge, l'université technologique d'Helsinki et l'institut Eurecom à Nice Sophia Antipolis. Il portait principalement sur les MANET. Ce projet de 3 ans a donné lieu à un grand nombre de publications (plus d'une vingtaine par an). Certains articles issus du projet portent sur l'optimisation inter-couches dans les MANET. En soit, le projet MobileMAN repose sur une optimisation sur toutes les couches [Con+04]. La Figure 3.4 illustre une architecture issue du projet. Le partage d'information s'effectue par une base de données partagée (*Network Status*).

Différentes implémentations d'optimisations inter-couches ont été réalisées sur des réseaux ad-hoc mobiles. Par exemple, Conti *et al.* [CGT05] ont modifié le protocole pair-à-pair Gnutella pour qu'il utilise les tables de routage de OSLR (*Optimized Link State Routing Protocol*, RFC 3626). D'après leur simulation réalisée avec *ns-2* avec un réseau IEEE 802.11, la version modifiée de Gnutella permet de meilleures performances. Par exemple, le surcoût moyen de communication est divisé par

2. En conditions réelles, Gnutella modifié permet un meilleur taux de requêtes réussies. Avec le protocole pair-à-pair Scribe, l'optimisation inter-couches permet de diminuer le délai entre chaque nœud et de réduire globalement le nombre de paquets perdus sur un réseau IEEE 802.11b [PDC06].

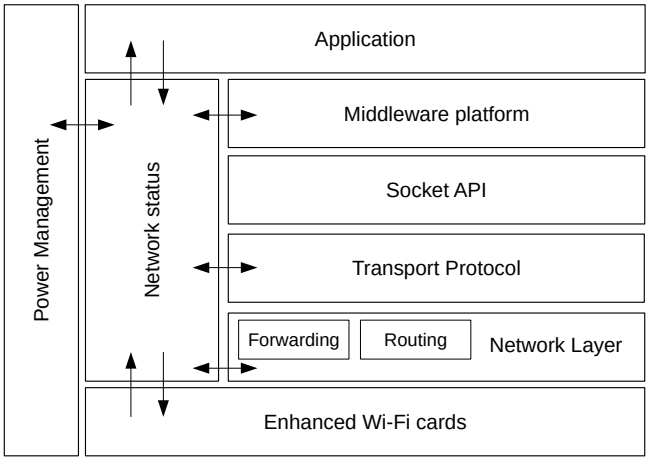


FIG. 3.4 : Architecture s'appuyant sur NeSt (*Network Status*) issue du projet MobileMAN, extrait de [BCD06]

Setton *et al.* [Set+05] proposent d'utiliser l'optimisation inter-couches pour transporter un flux vidéo en temps réel. Le réseau d'étude est un réseau IEEE 802.11 ad-hoc. Ils proposent d'effectuer une optimisation sur toute la pile protocolaire. L'objectif est que tout niveau de service sache quelle est la capacité du lien ou le débit disponible. En complément, le protocole de routage essaie d'utiliser les chemins les moins congestionnés. D'après leurs résultats, l'optimisation inter-couches permet de passer d'un débit de 250 kbit/s à 1,9 Mbit/s soit un débit multiplié par 7.

Réseaux industriels de capteurs

Les réseaux de capteurs partagent des caractéristiques des réseaux ad-hoc. Cependant, ils ont plus de contraintes sur le plan énergétique. Pour cette raison, ils sont rarement mobiles. Une partie des travaux sur les réseaux ad-hoc sont néanmoins réutilisables. Les réseaux de capteurs ont aussi fait l'objet de recherche sur de potentielles optimisations inter-couches. OCARI (*Optimization of Communications in an Ad hoc Reliable Industrial network*) est un projet français, incluant notamment l'Inria, de pile technologique pour les réseaux ad-hoc industriels et réseaux de capteurs [Ber+08]. OCARI se veut l'équivalent de ZigBee pour les réseaux industriels, c'est-à-dire une pile technologique sans-fil avec des forts couplages entre ses éléments. La Figure 3.5 montre les différents composants d'OCARI et leurs interactions. Des protocoles optimisés ont été créés durant le projet comme MaCARI, EOLSR et SERENA [Al +09].

MaCARI est le protocole de liaison du projet OCARI. D'après l'article [LBV08] sur sa conception, MaCARI permet une différentiation de service, un fonctionnement entièrement déterministe ou

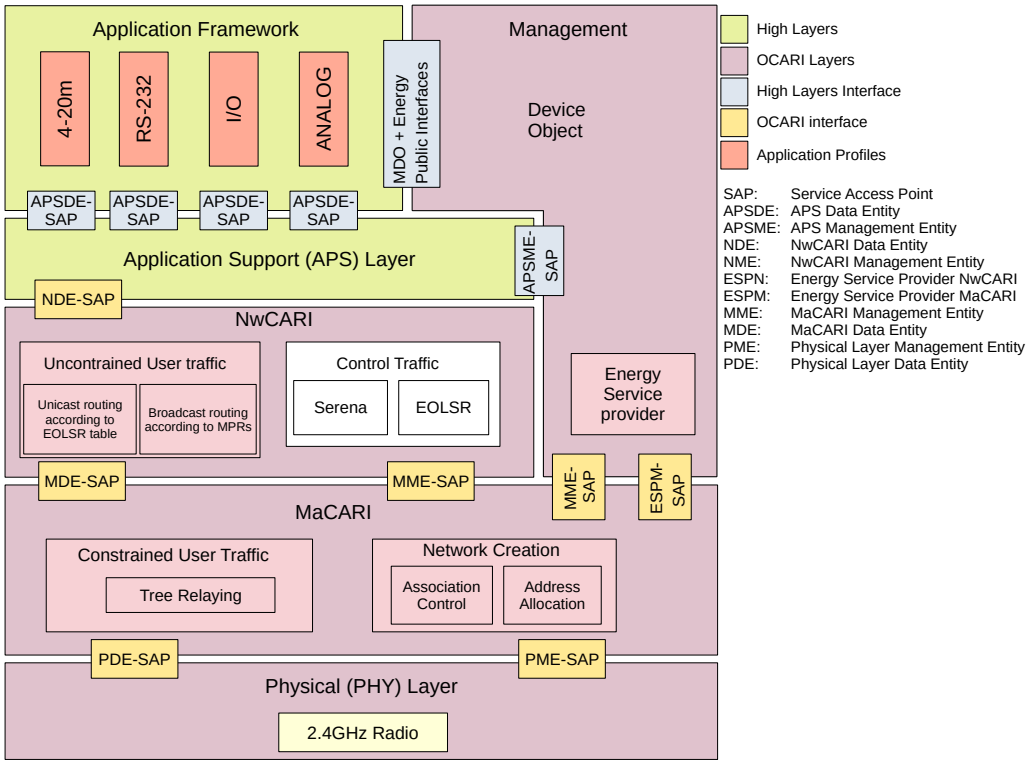


FIG. 3.5 : Architecture d'OCARI, extrait de [Ber+08]

une optimisation de bande passante. La topologie est de type arbre, il existe donc 3 rôles : PANC (*Personal Area Network Coordinator*, nœud racine), coordinateur (nœud intermédiaire) et nœud terminal. Les coordinateurs sont les seuls à avoir une capacité de routage. L'optimisation inter-couches s'effectue avec la couche réseau, plus précisément avec les protocoles de routage (protocole de routage hiérarchique et EOLSR). EOLSR est une version efficiente énergétiquement d'OLSR [MM11]. De nouvelles fonctionnalités ont été ajoutées. Par exemple, les paquets *HELLO* contiennent des informations sur l'énergie des nœuds afin de faire de l'équilibrage de charge. Une optimisation est effectuée avec MaCARI, la couche applicative basse ainsi que le module de gestion d'énergie. Par exemple, MaCARI fournit des informations sur la qualité du signal. Dans un réseau de capteurs, les nœuds sont en veille la plupart du temps. Il est donc possible qu'un nœud envoie des données à un nœud voisin en veille, les données sont alors perdues. Pour remédier à ce problème, un algorithme de planification a été créé. SERENA (*SchEdule RoutEr Nodes Activity*) permet d'organiser la veille et d'obtenir une meilleure efficience énergétique. D'après l'étude [MM09], SERENA permettrait de multiplier par 2 la durée de vie des équipements sur batteries. SERENA communique avec MaCARI et la couche applicative basse. Actuellement, moins d'une dizaine de modules de communication supportent la pile OCARI. En 2019, Carlotti *et al.* [Car+19] ont effectué des tests de réseaux de capteurs s'appuyant sur OCARI en conditions réelles. D'après ces tests, OCARI permet de diviser par 3

la consommation d'énergie par rapport à ZigBee.

3.3.3 Les limites de l'optimisation inter-couches

L'optimisation inter-couches n'a pas que des avantages. Nous allons maintenant voir quelques limites à l'optimisation inter-couches et plus généralement à la conception inter-couches.

Création de couplage et de dépendance inter-couches

Le concept de conception inter-couches est sujet à la création de couplage. Cela est dû notamment à la création d'interfaces. Les couches doivent avoir des interfaces compatibles pour pouvoir fonctionner correctement. Kawadia et Kumar [KK05] mettent en avant que la longévité des solutions avec conception inter-couches n'est pas garantie. Ils mettent en avant la standardisation des solutions et notamment des interfaces. Sur les exemples vus précédemment, la question de la pérennité n'a jamais été évoquée. Il y a eu des tentatives de standardisation IETF par exemple, mais elles n'ont pas abouti et sont restées à l'état de *draft* [Sar07 ; Wei12 ; Lee+11 ; CYC16].

Les contre-performances de conception inter-couches

La conception inter-couches a été créée pour simplifier la conception de protocole en s'affranchissant d'un certain nombre de contraintes. Il permet potentiellement d'obtenir de meilleures performances. Néanmoins, certaines conceptions inter-couches donnent des résultats moins bons que leur version originale. Kawadia et Kumar [KK05] proposent un exemple de conception inter-couches qui n'est volontairement pas une optimisation. Ils proposent d'utiliser IEEE 802.11b avec quelques modifications. Ils ajoutent un débit adaptatif au niveau de la couche liaison. Ils forcent aussi l'utilisation du chemin avec le nombre de sauts minimal. La simulation sur *ns-2* s'effectue sur un réseau ad-hoc de topologie linéaire et aléatoire. Dans les deux, IEEE 802.11b offre un meilleur débit que la proposition avec conception inter-couches. Le débit est environ deux fois plus élevé avec IEEE 802.11b. Le chemin le plus court n'est pas celui qui offre le meilleur débit. Une autre explication est liée aux interférences. Le chemin le plus court peut subir potentiellement plus d'interférences qu'un chemin avec plus de sauts.

3.4 Réseaux cognitifs

Dans cette dernière section, nous allons aborder les réseaux cognitifs. Nous verrons quelques architectures présentes dans la littérature, leur éventuelle implémentation et les performances de celles-ci.

Les prémices des réseaux cognitifs sont apparues en 2003. Clark *et al.* [Cla+03] proposent *Knowledge Plane*. *Knowledge Plane* est doté d'un système haut niveau de configuration de réseaux. Il est capable de reconfigurer automatiquement le réseau si besoin, de détecter des anomalies et réparer l'erreur automatiquement si possible. L'expression « réseaux cognitifs » est apparue une année plus

tard, en 2004. En mars 2004, Mähönen *et al.* [Mah+04] évoquent la possibilité d'étendre la radio cognitive définie par Mitola [Mit00]. Cette extension de fonctionnalités renommerait alors la radio cognitive en réseau cognitif. Par contre, Mähönen *et al.* ne définissent pas et ne développent pas le concept de réseaux cognitifs.

En mars 2004, Ramming [Ram04] définit les grandes lignes du concept de réseaux cognitifs. Il évoque les idées pour mettre en œuvre ces réseaux cognitifs. La première idée est de redéfinir la limite de responsabilités entre le réseau et les applications. En d'autres termes, il s'agit d'utiliser des optimisations inter-couches [1] pour que les applications puissent régler finement les différents paramètres réseaux. Une autre hypothèse étudiée est la création d'un langage de représentation comme RKRL (*Radio Knowledge Representation Language*) [MM99]. La dernière piste envisagée est l'utilisation de l'apprentissage automatisé. Les algorithmes bio-inspirés, la théorie des jeux et l'apprentissage par renforcement sont cités comme de potentiels candidats.

3.4.1 Définition d'un réseau cognitif

Thomas *et al.* [TDM05] sont les premiers à définir précisément le concept de réseaux cognitifs en 2005. Les réseaux cognitifs empruntent des éléments de la radio cognitive et de l'optimisation inter-couches. Les réseaux cognitifs étendent le concept de paramétrage et de reconfiguration de la radio cognitive. La radio cognitive fonctionne par définition uniquement avec les réseaux sans-fil. Les réseaux cognitifs ne sont pas limités par le type de réseaux. En effet, le concept de réseaux cognitifs peut fonctionner avec les réseaux sans-fil en mode ad-hoc et en mode infrastructure, mais aussi avec les réseaux filaires et les réseaux hétérogènes. De plus, les réseaux cognitifs ne sont pas limités aux couches basses du modèle OSI. Il est possible de modifier tous les paramètres de la pile protocolaire. Ce dernier point est commun avec l'optimisation inter-couches. Mais contrairement à la conception inter-couches, les réseaux cognitifs ne se limitent pas à optimiser un paramètre ou un indicateur de performances. Dans les réseaux cognitifs, un compromis est possible entre les différents éléments à optimiser. Un point crucial est que les optimisations se font à l'échelle du réseau, de bout-en-bout.

3.4.2 Quelques exemples d'applications des réseaux cognitifs

L'intérêt des réseaux cognitifs est en soi plutôt limité. Son principal intérêt est surtout d'améliorer la qualité de services d'un réseau. Les réseaux cognitifs peuvent améliorer plusieurs indicateurs de performances. Comme pour la radio cognitive, les réseaux cognitifs peuvent améliorer la fiabilité d'un réseau. Dans les réseaux cognitifs, les optimisations s'effectuent de bout-en-bout. Il est possible d'améliorer globalement la fiabilité. Le délai comme le débit peuvent être optimisés en choisissant les meilleures routes. À partir de simulations sur OPNET sur un réseau sans-fil IEEE 802.11b, Lee *et al.* [Lee+07b] montrent qu'en optimisant notamment le routage d'AODV, il est possible de diminuer la latence de 43 % et la gigue de 65 % par exemple. Ces deux métriques sont importantes pour la qualité d'expérience de certains services comme la VoIP et la vidéo conférence. Zorzi *et*

[1] Voir Section 3.3

al. [Zor+15] proposent d'utiliser les réseaux cognitifs pour la mise en place d'un contrôleur de trafic vidéo intelligent. Néanmoins, leur idée n'a pas été testée que ce soit en simulation ou en condition réelle. Un autre exemple consisterait à utiliser les réseaux cognitifs pour améliorer la sécurité. En effet, les réseaux cognitifs ont déjà accès au statistique du réseau. Il serait possible de détecter des anomalies dans ces conditions.

3.4.3 Architecture, implémentations et résultats

Nous allons aborder quelques architectures de réseaux cognitifs. Nous verrons leur éventuelle implémentation ainsi que les performances associées. Chaque sous-sous-section présente une architecture. Plusieurs architectures se nomment « cognet ». La référence à « cognet » réfère par défaut à l'architecture de la sous-sous-section.

Cognitive Framework

Thomas *et al.* [TDM05 ; Tho+06] ont proposé une architecture pour les réseaux cognitifs. La Figure 3.6 représente l'architecture de ce cadre. Il se compose de 3 parties : le réseau logiciel adaptable (*Software adaptable network*), une partie concentrant l'Intelligence et enfin une partie définissant les objectifs. Cette architecture possède plusieurs points critiques de conception. Le premier point relève de la capacité à modifier les éléments du réseau. Il est possible d'avoir une emprise complète ou partielle sur la configuration. En lien avec ce point, le deuxième point critique concerne le comportement des nœuds. Les nœuds peuvent agir de manière égoïste ou collaborativement. Ces deux modes ont un coût de fonctionnement différent. Par exemple, en mode égoïste, les nœuds vont essayer de satisfaire leurs propres objectifs aux détriments des performances globales. La différence entre le fonctionnement égoïste et le fonctionnement optimal porte le nom de « prix de l'anarchie » [Pap01]. L'avantage du mode égoïste est qu'il permet une meilleure robustesse [Rou09]. La coopération a aussi un coût lié à la communication entre les nœuds (« prix du contrôle »). Le dernier point concerne le partage de connaissance entre les différents éléments du réseau. Par exemple, la remontée d'une grande quantité d'informations aux nœuds intelligents permet une meilleure précision dans les décisions. Néanmoins, ces informations ont un coût en termes de calcul et de bande passante [Kom+10].

Management automatique cross-layer

En 2007, Lee *et al.* [Lee+07b] proposent d'utiliser l'apprentissage automatique pour le management automatique de réseaux maillés. La Figure 3.7 illustre cette architecture. Ce système peut fonctionner avec les réseaux ad-hoc et les réseaux hétérogènes. Ils proposent en particulier d'utiliser l'apprentissage par renforcement et plus particulière la technique Q-Learning. L'avantage du Q-Learning est qu'il ne nécessite pas de modèle de départ. Dans ce cas, Q-learning utilise le réseau comme environnement d'apprentissage. L'état du réseau est représenté par un ensemble de métriques extraites du réseau.

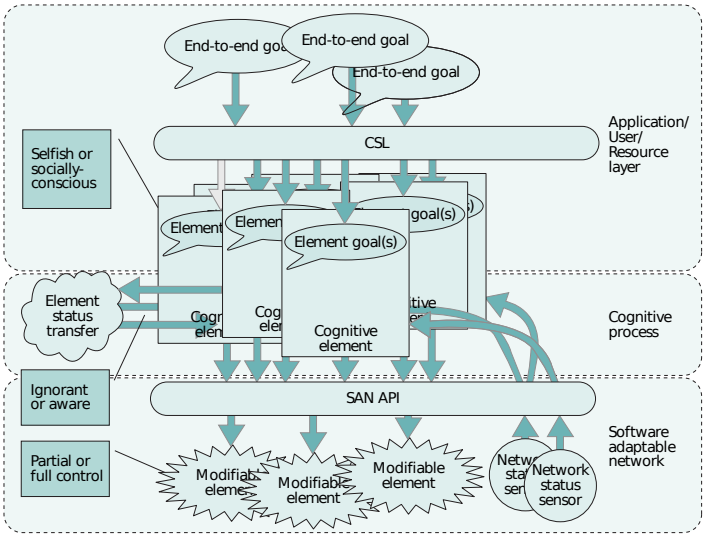


FIG. 3.6 : Architecture du *Cognitive Framework* extrait de [Tho+06]

Des simulations ont été réalisées avec *OPNET*. Le réseau simulé est de type MANET. Les nœuds communiquent en IEEE 802.11b. La simulation consiste à évaluer la différence de performances en AODV et AODV configuré dynamiquement par Q-learning. Le trafic est du flux vidéo. Trois situations sont évaluées : MANET à MANET, MANET à réseau filaire et inversement. Les métriques d'évaluation sont : la durée de découverte d'une route, le nombre d'erreurs de route, le surcoût lié au contrôle de topologie, la gigue, le délai de bout-en-bout et le ratio de paquets délivrés. AODV renforcé offre de meilleures performances. Par exemple, le taux de paquet délivré augmente de 14 %. Le délai de bout-en-bout est en moyenne 43 % plus court.

Cognitive networking framework et CogNet AP

Pour Manoj *et al.* [MRZ07a], la mise en place d'une couche physique intelligente est trop coûteuse et complexe par rapport au gain. Ils proposent de n'utiliser que les informations des couches hautes. Ils pensent que ces informations sont suffisantes pour la mise en place d'un réseau cognitif. L'architecture proposée par Manoj *et al.* ne respecte pas leur affirmation concernant la couche physique. Leur architecture est distribuable et implique un surcoût dû aux communications inter-nœuds. La Figure 3.8 illustre leur architecture. La partie intelligente est nommée CogPlane. Elle est composée de deux éléments : CogBus et Cognitive Executive Function (CEF). Les données de chaque couche sont récupérées par une abstraction (Cognitive Agent, CA) reliée au bus inter-couches (CogBus). CEF est une optimisation *joint layer* distribuée. Nous avons vu dans la section 3.3 que les *joint layer design* étaient des optimisations très difficiles à mettre en œuvre.

En dehors de leur architecture, Manoj *et al.* proposent aussi CogNet AP (*Access Point*). L'idée derrière CogNet AP est de créer un « réseau cognitif » autonome à l'échelle du point d'accès. La no-

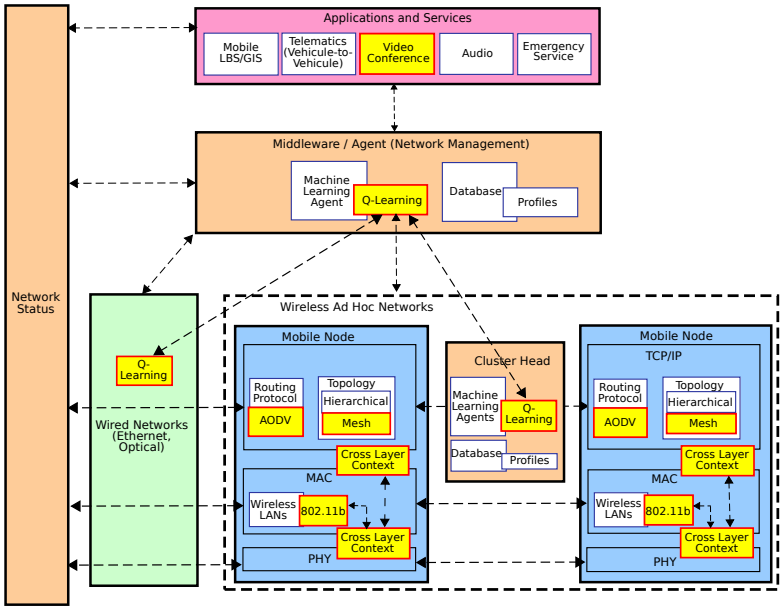


FIG. 3.7 : Architecture de réseau cognitif avec prise de décision distribuée extrait de [Lee+07a]

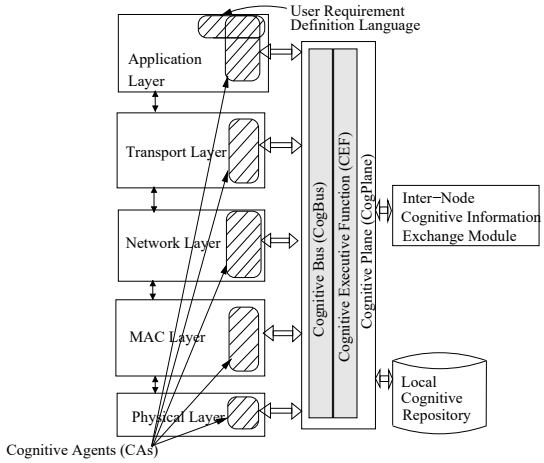


FIG. 3.8 : Architecture Cognitive networking framework extrait de [MRZ07a]

tion d'optimisation de bout en bout n'est donc plus disponible. La Figure 3.10 illustre l'architecture du CogNet AP. Une des antennes du point d'accès est utilisée activement pour le réseau. L'autre antenne est en écoute passive. L'antenne en écoute capture toutes les trames sur le canal utilisé. Le changement de canal s'effectue toutes les minutes afin de balayer tous les canaux disponibles. Les données sur des couches supérieures sont extraites à partir des trames capturées. Des statistiques sont ensuite calculées et stockées à partir de ces données. Les données collectées sont la durée

moyenne inter-arrivées des paquets IP et pour chaque protocole applicatif, la longueur moyenne des paquets IP et pour chaque protocole applicatif reçus, le nombre de paquets IP et chaque protocole applicatif. En complément, une étiquette temporelle est notamment ajoutée. CogNet AP choisit automatiquement le canal en fonction des données stockées. En raison de sa portée d'action et les actions réalisées, CogNet AP est plus une implémentation de radio cognitive que de réseau cognitif. L'implémentation CogNet AP présentée est considérée par Manoj *et al.* comme un premier prototype en termes de fonctionnalités.

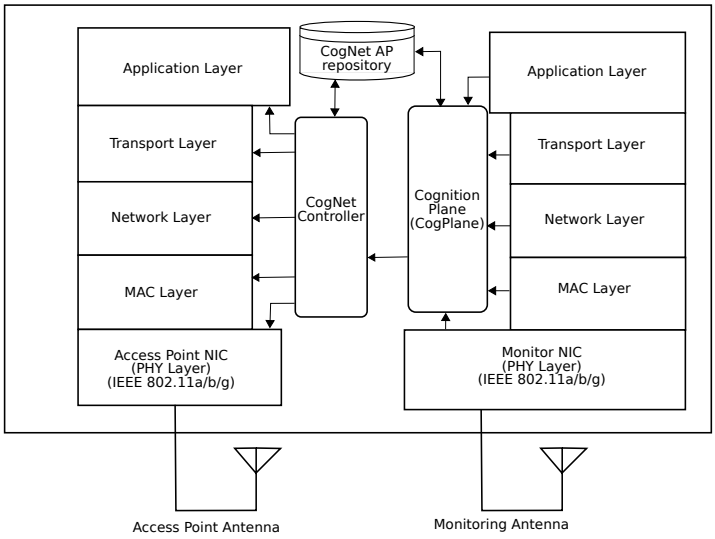
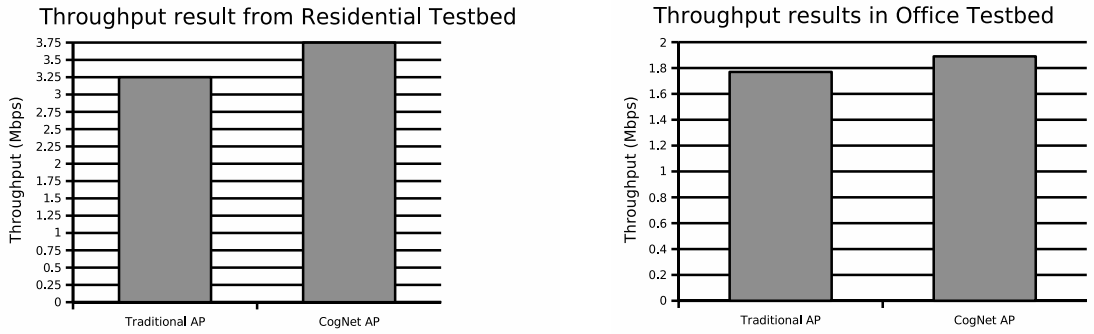


FIG. 3.9 : Architecture du CogNet AP extrait de [MRZ07a]

CogNet peut être implémenté avec du matériel sur étagère. Manoj *et al.* ont implémenté CogNet AP et ont effectué quelques tests. Ces tests ont été effectués dans deux environnements différents. Le premier environnement est un appartement dans une zone résidentielle. Le second environnement est composé de bureaux. Avec 2 à 3 semaines d'informations, le débit a augmenté jusqu'à 10 % en fonction de l'environnement. Les figures 3.10 illustrent ces résultats pour les deux environnements précités. Le très faible gain de bande passante de CogNet AP dans les environnements de bureaux peut s'expliquer par une plus grande densité des réseaux sans-fil. CogNet AP peut plus difficilement choisir des canaux libres.

Cognitive Complete Knowledge Network

Nous allons voir maintenant l'architecture Cognitive Complete Knowledge Network de Manoj *et al.* [MRZ08]. La Figure 3.11 illustre l'architecture de Cognitive Complete Knowledge Network (abrégé aussi CogNet). Cette architecture est principalement sur les réseaux sans-fil. Elle est proche de la radio cognitive. CogNet est divisé en 3 parties : la plateforme radio logicielle, le réseau et la partie « contrôle et intelligence ». Il est possible de modifier uniquement les paramètres de la couche



(a) en environnement « résidentiel »

(b) en environnement de bureaux

FIG. 3.10 : Résultats des tests de débit avec CogNet AP extrait de [MRZ07a]

physique et liaison. Des informations issues de toutes les couches sont néanmoins collectées. Pour cela, CogNet utilise des abstractions sur toutes les couches. Les abstractions sont reliées entre elles et au reste du système grâce à un bus commun.

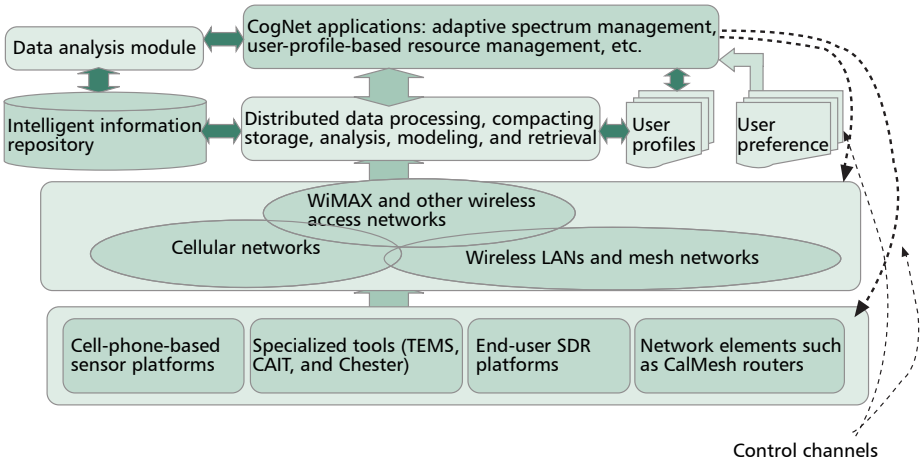


FIG. 3.11 : Architecture Cognitive Complete Knowledge Network extrait de [MRZ08]

Dans ce contexte, Manoj *et al.* [MRZ08] proposent CogTCP, une version de TCP utilisant l'architecture CogNet. L'évaluation de CogTCP a été réalisée sur le simulateur *GloMoSim*. La topologie est de type grille. IEEE 802.11 à 2 Mbit/s est utilisé pour la communication entre les nœuds. Un serveur FTP est disposé dans un coin de la grille, deux nœuds clients (A et B) sont placés aux coins opposés. Le nœud A va récupérer un fichier sur le serveur, tous les paramètres seront tracés et stockés. Le nœud B va ensuite récupérer le fichier depuis le serveur. Lorsqu'il utilisera CogTCP, il pourra profiter de l'expérience acquise avec le nœud A. Avec CogTCP-1, le nœud B utilise la valeur moyenne de la taille de fenêtre de congestion TCP comme valeur initiale. Avec CogTCP-2, il prend la moitié de la

valeur moyenne de la taille de fenêtre de congestion comme valeur initiale. Deux métriques servent à évaluer les deux versions de CogTCP : la taille de la fenêtre de congestion ainsi que le débit. Le tableau 3.1 montrent les différences de débit entre TCP et les deux versions de CogTCP. Les deux versions de CogTCP améliorent légèrement le débit (jusqu'à 10 % dans 3 configurations). La taille de la fenêtre de congestion est en moyenne plus grande qu'avec TCP. Le gain est le plus important lorsque les fichiers à transférer sont de petite taille car la valeur maximum de la taille de la fenêtre de congestion n'est pas atteinte. Pour les fichiers volumineux, la taille de fenêtre maximum est atteinte plusieurs fois en cours du transfert. Dans ces cas, la taille de fenêtre est divisée par 2, ce qui ralentit le transfert et explique que le gain est minime. Les résultats de CogTCP-2 sont très légèrement meilleurs que ceux de CogTCP-1 car CogTCP-2 atteint moins rapidement la taille maximum de la fenêtre.

Débit sans trafic en arrière-plan (kb/s)

	TCP	CogTCP-1	CogTCP-2
Sessions de transfert de fichier courtes (10 ko)	127	139	139
Sessions de transfert de fichier longues (1 Mo)	62	66	68

Débit avec trafic en arrière-plan (kb/s)

	TCP	CogTCP-1	CogTCP-2
Sessions de transfert de fichier courtes (10 ko)	120	135	135
Sessions de transfert de fichier longues (1 Mo)	55	57	58
Sessions de transfert de fichier longues (10 Mo)	44	50	50

TAB. 3.1 : Évaluation des performances des deux versions de CogTCP par rapport TCP, [MRZ07b]

CARMEN

Cognitive Android Mesh Network (CARMEN) est une plateforme d'expérimentation pour les réseaux cognitifs [Dan+14]. Elle se distingue des autres plateformes radio logicielle par le faible coût des terminaux utilisés. En effet, CARMEN est composée de tablette et de smartphone de grand constructeur fonctionnant avec le système Android. Le faible prix des terminaux n'implique pas une diminution drastique des fonctionnalités. Ils peuvent communiquer aussi bien en Wifi (IEEE 802.11b/g/n) et qu'en Bluetooth (BT 4.0). Il est possible de fonctionner en mode ad-hoc, de créer des topologies de type maillage. Par leur nature, il est facile d'utiliser la plateforme dans des scénarios nécessitant la mobilité des nœuds. Un certain nombre de paramètres sont accessibles en fonction des pilotes réseaux. Dans les meilleurs des cas, 8 paramètres (4 TCP, 4 MAC) sont disponibles en lecture/écriture. Par exemple, il est possible de modifier la valeur de fenêtre de contention TCP, l'espace arbitraire inter-trame (*Arbitration Inter-Frame Spacing*, AIFS). 25 autres paramètres (12 TCP, 13 MAC) sont accessibles en lecture seulement. Afin de gérer plus facilement ces paramètres ainsi que les paramètres de routage, un ensemble de composants logiciels a été développé. Cet ensemble s'appelle encore Cognet. Il est composé de trois parties : CognetModule, CognetNode et Cognet-

Manager. La Figure 3.12 illustre l'architecture de Cognet. Une plateforme comme CARMEN serait plus difficile à mettre en place aujourd'hui. En effet, pour mettre en place les différents éléments, il faut notamment les droits d'administration avancée ainsi qu'un chargeur d'amorçage déverrouillé (premier élément logiciel au démarrage du système d'exploitation). Ces deux prérequis sont de plus en plus difficiles à valider, car ils reposent sur des éléments de sécurité. D'autres détails techniques compliquent aussi la mise en place d'une telle plateforme.

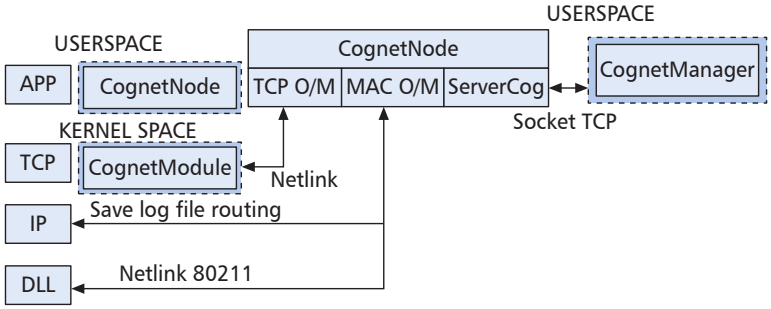


FIG. 3.12 : Architecture de *Cognet* extrait de [Dan+14].

CogNet de Xu *et al.*

Dans le cadre du projet européen CogNet impliquant notamment Orange, IBM et l'Université Polytechnique de Madrid lié aux réseaux de télécommunication de cinquième génération (5G), Xu *et al.* [Xu+16] proposent d'utiliser le concept de réseaux cognitifs. Les réseaux cognitifs sont utilisés afin d'obtenir une infrastructure auto-gérée, capable de compromis entre la qualité de service, la consommation énergétique et l'efficience opérationnelle. La Figure 3.13 illustre l'architecture résultant de leurs travaux. Leur architecture s'appuie sur les réseaux logiciels et la virtualisation de fonctions réseaux (*Network Function Virtualisation*, NFV). CogNet intègre les techniques issues du *Big Data* pour le traitement des données. Il intègre notamment l'apprentissage automatique pour réaliser l'extraction des caractéristiques ainsi que pour prendre des décisions. L'architecture a été conçue pour traiter les données en quasi-temps réel.

3.4.4 Conclusions sur les réseaux cognitifs

Nous avons vu plusieurs architectures de réseaux cognitifs. Il ne s'agit que de quelques exemples. Malheureusement, très peu de travaux ont dépassé l'étape de l'architecture. Sur les six réseaux cognitifs proposés, seul Cognet de Danieletto *et al.*[Dan+14] a été implémenté. Bien que leur implémentation soit fonctionnelle sur du matériel sur étagère, elle n'a jamais été pleinement exploitée. En effet, dans leur expérimentation, Danieletto *et al.* ne changeaient qu'un seul paramètre à la fois, loin de l'optimisation globale de paramètres. Malheureusement, leur implémentation est maintenant obsolète. Comme pour les optimisations inter-couches, les réseaux cognitifs souffrent de difficulté de pérennité. Quelques travaux comme [MRZ07a ; MRZ07b ; Lee+07a] ont réduits leurs

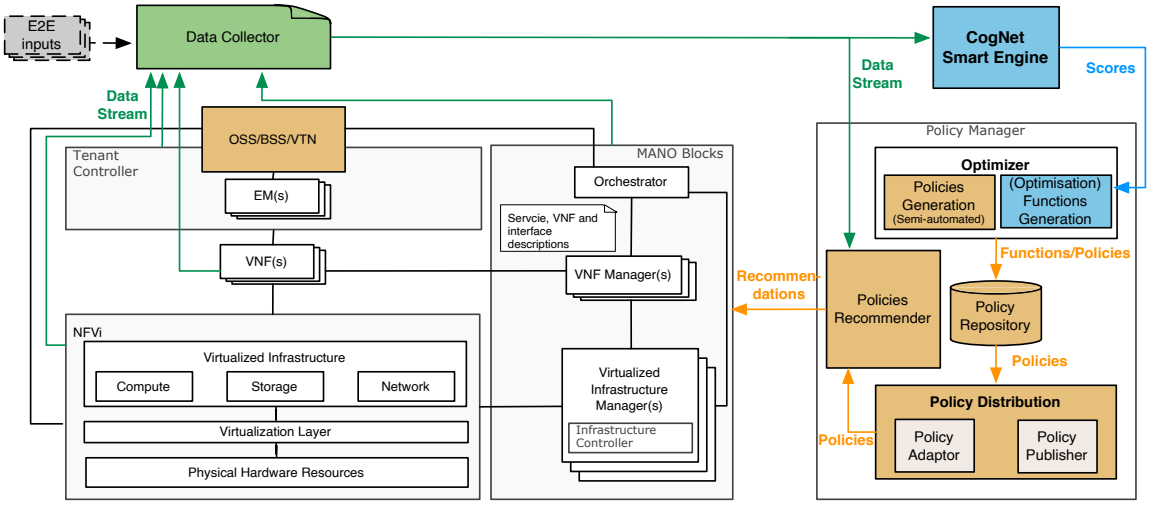


FIG. 3.13 : Architecture Cognet de Xu *et al.* extrait de [Xu+16]

ambitions pour n'implémenter qu'une seule fonctionnalité de leur architecture. Dans leur cas, les résultats obtenus sont encourageants.

3.5 Conclusions

Dans ce chapitre, nous avons abordé deux types d'interférences et leurs effets destructeurs notamment sur les réseaux sans-fil. Ce problème est récurrent car la plupart des technologies des réseaux sans-fil utilisent les mêmes bandes de fréquences : les bandes de fréquences ISM 868 MHz et 2,4 GHz. Cette situation favorise la création d'interférences. Nous avons relevé quelques solutions matérielles et logicielles afin d'atténuer le problème. Au-delà des interférences, il est possible d'améliorer les réseaux sans-fil. Nous avons abordé deux méthodes : les optimisations inter-couches et les réseaux cognitifs. Chacune de ces techniques a ses inconvénients. Par exemple, les optimisations inter-couches cassent la modélisation OSI. Elles deviennent donc plus difficiles à utiliser dans des cas généraux. Même si les gains de performances espérés sont réduits, il existe plusieurs implémentations « matérielles » de conception inter-couches. Les réseaux cognitifs sont en théorie les plus performants, mais ils ont rarement été mis en œuvre. Les auteurs ont généralement revu leurs ambitions à la baisse en implémentant qu'une seule fonctionnalité. Néanmoins, l'amélioration d'une seule fonctionnalité permet déjà d'obtenir de meilleures performances. Ces améliorations sont obtenues grâce à de l'apprentissage automatique. Nous ne pensons pas intéressant de proposer une autre architecture. De plus, l'exercice ne présente pas d'intérêt particulier dans le cadre du projet COWIN. La réimplémentation de Cognet de [Dan+14] sur une autre plateforme peut sembler une idée séduisante. Cependant, la réalisation de l'implémentation est soumis à de fortes contraintes techniques comme le signale Danieletto *et al*. De plus, l'implémentation est chronophage, sans pour autant qu'il y ait finalement un quelconque gain. Dans cette thèse, nous allons donc nous

concentrer sur l'amélioration d'une fonctionnalité. Dans le prochain chapitre, nous aborderons les améliorations de fonctions réseau réalisées grâce à l'apprentissage par renforcement.

Chapitre 4

L'apprentissage par renforcement appliqué au routage

4.1 Introduction

Les problèmes des réseaux informatiques sont complexes, que ce soit le routage, la classification du trafic ou la prédiction de pannes. L'apprentissage automatique peut permettre de résoudre ces problèmes. L'apprentissage automatique est composé de plusieurs catégories. Les trois catégories principales sont l'apprentissage supervisé, non supervisé et par renforcement. Dans le cas de l'apprentissage supervisé, l'objectif est de créer un modèle à partir de données labellisées. L'apprentissage supervisé peut être utilisé pour effectuer de la classification de trafic [Pac+19]. Dans le cas de l'apprentissage non supervisé, l'objectif est de créer un modèle à partir de données non labellisées. Il peut être utilisé par exemple pour la détection d'anomalies [Fal+19]. Contrairement aux deux autres approches, le modèle est créé par essais et erreurs (*try and error*) pour l'apprentissage par renforcement. Il ne nécessite donc pas de jeu de données initial d'entraînement, ce qui est un atout. Les problèmes de routage peuvent être résolus grâce à l'apprentissage par renforcement comme nous allons le voir par la suite. Nous allons commencer par présenter plus en détail l'apprentissage par renforcement (Section 4.2). Nous verrons ensuite des travaux utilisant l'apprentissage par renforcement pour résoudre des problèmes des réseaux informatiques. Enfin, nous terminerons par Q-routing [BL94], un algorithme de routage inspiré de l'algorithme d'apprentissage par renforcement Q-learning [WD92] (Section 4.4).

4.2 Apprentissage par renforcement

Nous allons commencer par définir un peu plus en détails l'apprentissage par renforcement. L'apprentissage par renforcement est le troisième paradigme d'apprentissage automatique. Contrairement à l'apprentissage supervisé et non supervisé, il ne nécessite pas nécessairement de jeu de données initial. L'apprentissage s'effectue à partir de ses expériences (*try-and-error*).

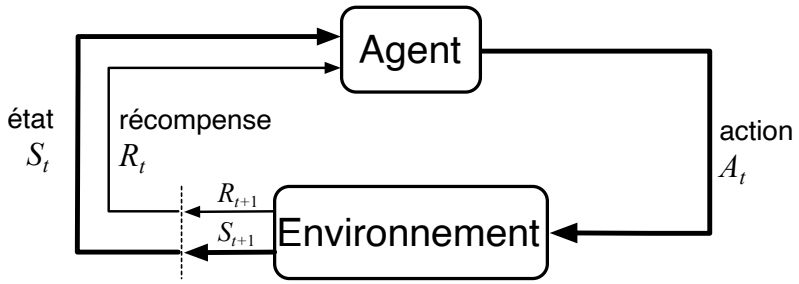


FIG. 4.1 : Interaction entre l'agent et l'environnement dans un processus de décision Markovien, extrait de [SB18].

L'apprentissage par renforcement définit quatre notions fondamentales : l'agent, l'environnement, l'action et la récompense. Les interactions entre ces quatre notions sont illustrées par la Figure 4.1 qui résume le cadre d'un processus de décision Markovien (MDP). De manière pédagogique, nous allons utiliser une analogie d'un joueur dans une salle remplie de machines à sous [SB18]. Ce problème est aussi connu sous le nom de bandit manchot. Le joueur est notre agent. La salle remplie de machines à sous est notre environnement. Les machines à sous composent l'ensemble des états de l'environnement. Une action consiste à jouer sur une des machines à sous. La récompense est le gain obtenu. Chaque machine a une espérance de gain qui lui est propre. L'objectif dans ce cas consiste à maximiser les gains cumulés soit à long terme, soit sur une période (en nombre d'essais).

L'apprentissage par renforcement définit quatre notions supplémentaires : la fonction de valeur, la stratégie, le signal de récompense et le modèle. Nous reprenons notre analogie précédente. Comme l'espérance de chaque machine n'est pas connue, elle peut être au mieux estimée. La fonction de valeur est le nom de la fonction associant une machine à sous et l'estimation de son espérance. Le processus de choix de la machine à sous compose la stratégie de l'agent. Par exemple, la stratégie gloutonne (*greedy*) consisterait à toujours choisir la machine à sous dont l'espérance estimée est la plus élevée. Le signal de récompense est l'objectif de gain à réaliser. Le modèle permet d'effectuer des prédictions sur le comportement de l'environnement, notamment ses réactions par rapport à une action et l'état suivant. Certains algorithmes comme Q-learning ne nécessitent pas de modèle de l'environnement.

Dans une stratégie, il y a deux phases possibles : la phase d'exploration et la phase d'exploitation. Ces deux phases ont une finalité différente. La phase d'exploration sert à améliorer l'estimation des récompenses des différentes actions. L'objectif de la phase d'exploitation est uniquement d'atteindre l'objectif. Nous reprenons notre exemple de bandit manchot. Durant la phase d'exploration, le joueur va essayer une ou plusieurs machines à sous afin d'améliorer l'estimation de l'espérance des gains de celles-ci. Durant la phase d'exploitation, le joueur va utiliser la machine à sous permettant de maximiser les gains. Le passage d'une phase à l'autre est dépendant de la stratégie. Une stratégie gloutonne n'est composée que d'une phase d'exploitation.

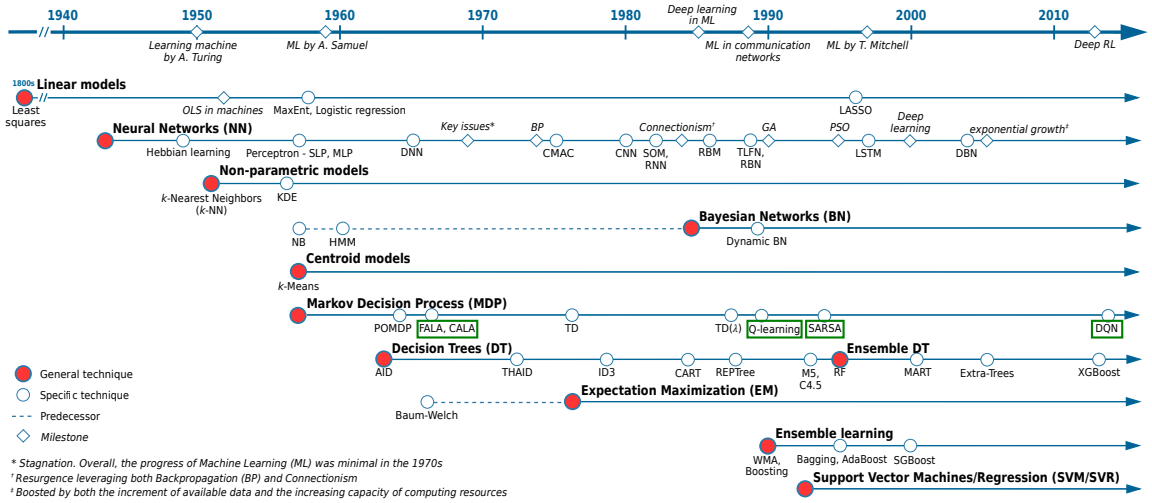


FIG. 4.2 : Développement de l'apprentissage automatique extrait de [Bou+18]. Les algorithmes cités dans ce chapitre sont encadrés en vert.

4.2.1 Quelques exemples d'algorithmes d'apprentissage par renforcement

Nous allons nous intéresser à quelques algorithmes d'apprentissage par renforcement dont Q-learning et ses dérivés. La modélisation d'un réseau informatique est très compliqué, nous avons choisi Q-learning car il ne nécessite pas de modèle, ni de jeu de données initial. La Figure 4.2 illustre le développement de l'apprentissage automatique ci-dessus. Tous les algorithmes présentés ou cités sont encadrés en vert. Ils appartiennent à la catégorie des processus de décision Markovien.

Q-learning

Q-learning est un algorithme d'apprentissage par renforcement créé pendant la thèse de doctorat de Watkins en 1989 et présenté en 1992 [WD92]. Q-learning définit A l'ensemble des actions a, S l'ensemble des états s de l'environnement et R l'ensemble des récompenses r. Afin de contrôler l'apprentissage, Q-learning utilise deux paramètres : le taux d'apprentissage α et le facteur dégressif γ. Le taux d'apprentissage α permet de pondérer la récompense actuelle. Plus la valeur du taux d'apprentissage α est grande, plus Q-learning va donner de l'importance aux dernières récompenses obtenues. Le facteur dégressif γ pondère l'influence de l'état suivant dans le calcul de la récompense de l'état actuel. Plus le facteur dégressif est proche de 1, plus le choix de Q-learning sera influencé par le prochain état. En pratique, le prochain état n'est pas toujours connu. Dans ce cas, le facteur dégressif est nul. Enfin, la Q-fonction est la fonction d'association action-valeur. Il s'agit d'une extension de la fonction de valeur. La Q-fonction associe un état et une action à l'estimation de l'espérance des récompenses. Q-learning définit la Q-fonction telle que :

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right] \tag{4.1}$$

où A_t représente l'action sélectionnée à l'instant t, S_t l'état à l'instant t et R_t la récompense de l'action A_t à l'état S_t.

Algorithme 1 : Algorithme de Q-learning, tel que présenté dans [SB18]

Initialiser $Q(s, a)$ pour tout état $s \in S$, toute action $a \in A$ de façon arbitraire, sauf $Q(\text{état terminal}, \cdot) = 0$;
répéter
  Initialiser S ;
  répéter
    Choisir A depuis S d'après la stratégie et Q ;
    Exécuter A ;
    Observer R et S' ;
    $Q(S, A) \leftarrow Q(S, A) + \alpha\,[R + \gamma \max_a Q(s', a) - Q(S, A)]$;
    $S \leftarrow S'$;
  jusqu'à S *soit l'état terminal* ;
jusqu'à *fin* ;

La fonction d'association action-valeur optimale est notée q_*. q_* permettrait de choisir les actions optimales en fonction de chaque état si elle était connue. La Q-fonction approxime directement q_*. Son fonctionnement est décrit par l'Algorithme 1. Le principal avantage de Q-learning est qu'il ne nécessite pas de modèle. Un des inconvénients de Q-learning réside dans sa surestimation des récompenses (*maximization bias*). De fait, Q-learning va prendre plus de risques.

SARSA

SARSA (*State–Action–Reward–State–Action*) est un algorithme d'apprentissage par renforcement créé par Rummery et Niranjan en 1994 [RN94]. Il s'inspire de Q-learning. SARSA définit la Q-fonction par :

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\,[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \tag{4.2}$$

Son fonctionnement est décrit par l'Algorithme 2. Contrairement à Q-learning, SARSA exécute deux actions pour une mise à jour. SARSA a l'avantage de prendre moins de risques que Q-learning. Néanmoins, SARSA nécessite un modèle, ce qui le rend plus difficile à utiliser que Q-learning.

Double Q-Learning

Le Double Q-learning est un algorithme d'apprentissage créé en 2010 par Hasselt [Has10]. L'intérêt du Double Q-learning est qu'il ne surestime pas les récompenses contrairement à Q-learning. Cette surestimation conduit généralement Q-learning à choisir de mauvaises actions. Pour cela, Double Q-learning utilise deux estimateurs Q : Q^A et Q^B. Similairement à Q-learning, $Q^A \approx Q^B \approx$

Algorithme 2 : Algorithme de SARSA, tel que présenté dans [SB18]

Initialiser $Q(s, a)$ pour tout état $s \in S$, toute action $a \in A$ de façon arbitraire, sauf $Q(\text{état terminal}, \cdot) = 0$;
répéter
  Initialiser S ;
  répéter
    Choisir A depuis S d'après la stratégie et Q ;
    Exécuter A ;
    Observer R et S' ;
    Choisir A' depuis S' d'après la stratégie et Q ;
    $Q(S, A) \leftarrow Q(S, A) + \alpha\,[R + \gamma Q(S', A') - Q(S, A)]$;
    $S \leftarrow S'$;
    $A \leftarrow A'$;
  jusqu'à S *soit l'état terminal* ;
jusqu'à *fin* ;

q_*. Ces deux estimateurs ne sont pas mis à jour en même temps, comme décrit dans l'Algorithme 3. Les résultats de Double Q-learning sont donc meilleurs que ceux de Q-learning. Le principal inconvénient reste toutefois qu'il a besoin de deux fois plus de mémoire. En effet, il est nécessaire de stocker le double des triplets action, état, récompense.

Algorithme 3 : Algorithme de Double Q-learning, tel que présenté dans [Has10]

Initialiser $Q^A(s, a)$, $Q^B(s, a)$ et s ;
répéter
  Choisir une action a d'après $Q^A(s, \cdot)$ et $Q^B(s, \cdot)$;
  Exécuter a ;
  Observer la récompense r et l'état s' ;
  Choisir entre MIS_A_JOUR(A) et MIS_A_JOUR(B) ;
  si *MIS_A_JOUR(A)* **alors**
    Définir a^* = arg max$_a Q^A(s', a)$;
    $Q^A(s, a) \leftarrow Q^A(s, a) + \alpha(s, a)(r + \gamma Q^B(s', a^*) - Q^A(s, a))$;
  sinon
    Définir b^* = arg max$_a Q^B(s', a)$;
    $Q^B(s, a) \leftarrow Q^B(s, a) + \alpha(s, a)(r + \gamma Q^A(s', b^*) - Q^B(s, a))$;
  fin
  $s \leftarrow s'$;
jusqu'à *fin* ;

Apprentissage par renforcement profond

Comme pour les autres paradigmes d'apprentissage automatique, il existe de l'apprentissage par renforcement profond. Toutefois, il n'apparait qu'en 2015, soit plus de 20 ans après les autres paradigmes d'apprentissage. DQN (*Deep Q-Network*) est le premier algorithme d'apprentissage par renforcement profond [Mni+15]. DQN combine Q-learning et un réseau de neurones profond. Concrètement, le réseau de neurones profond remplace la Q-fonction. Il permet notamment la gestion d'un grand nombre d'entrées, comme une image brute par exemple. Les tâches réalisables peuvent être plus complexes, mais au prix d'une complexité accrue par rapport à Q-learning. Double DQN est une variante de DQN utilisant Double Q-learning [HGS16]. Ces deux algorithmes se sont illustrés par leur utilisation dans l'intelligence artificielle AlphaGo (pour le jeu de Go) et AlphaStar (pour le jeu vidéo de stratégie en temps réel StarCraft II) [Sil+17].

4.3 Applications de l'apprentissage par renforcement aux problèmes de réseaux informatiques

Dans cette section, nous allons voir deux exemples d'application d'apprentissage par renforcement pour les réseaux informatiques. La première utilisation concerne le paramétrage du protocole TCP. La fenêtre de contention de TCP influe beaucoup sur ses performances. L'idée principale est d'utiliser l'apprentissage par renforcement pour régler ce paramètre. La seconde utilisation concerne la fonction de routage car il s'agit d'une fonction complexe. En fonction de la topologie, le plus court chemin n'est pas le meilleur. L'apprentissage par renforcement permet notamment d'utiliser des chemins alternatifs ou d'équilibrer la charge entre différents liens.

4.3.1 Application au paramétrage automatique du protocole TCP

TCP est un protocole de transport orienté connexion. Son fonctionnement est paramétrable. Un de ces paramètres est la fenêtre de contention. La gestion de cette fenêtre fait partie des problèmes complexes du domaine des réseaux. Il existe de nombreuses stratégies pour gérer cette fenêtre. Par exemple, il existe New Reno (RFC 6582), CUBIC et Vegas. Chaque stratégie définit l'incrémentation de ce paramètre ainsi que sa valeur en cas de perte de segment TCP.

Afin de résoudre le problème d'optimalité de la valeur de la fenêtre de contention, Venkata Ramana *et al.* proposent d'utiliser l'apprentissage par renforcement avec TCP. Ils nomment leur proposition Learning-TCP [VMS05]. Learning-TCP s'appuie en grande partie sur TCP Reno. Pour la partie apprentissage, Venkata Ramana *et al.* utilisent FALA (*Finite Action-set Learning Automata*). D'après leurs simulations sur *GloMoSim*, Learning-TCP permet une amélioration globale des performances, que ce soit en termes de délai de livraison ou de taux de perte. En 2011, ils proposent une nouvelle version Learning-TCP [BM11]. Cette fois, la partie apprentissage est assurée par CALA (*Continuous Action-set Learning Automata*). D'après leurs simulations sur *ns-2*, cette nouvelle version de Learning-TCP permet d'obtenir de meilleures performances en termes de taux de paquets perdus et de débit que la première version de Learning-TCP et TCP New Reno.

En 2016, Li *et al.* [Li+16] proposent TCPLearning, un variant de TCP utilisant Q-learning. Ils ont implémenté leur algorithme sur le simulateur réseau *ns-3*. Il est comparé à TCP New Reno et à d'autres variants de TCPLearning. Leur topologie de test est très simple. Leur scénario est composé d'une source, d'une destination et d'un goulot d'étranglement. Ce nœud intermédiaire autorise en fonction du scénario une bande passante soit de 7,5 Mb/s, soit de 2,5 Mb/s. Dans le premier cas, TCPLearning permet d'améliorer le débit disponible de 26 % par rapport TCP New Reno pour un RTT (*Round Trip Time*) équivalent. Dans le second cas, il offre une diminution du RTT de 12 % pour une bande passante utile équivalente. La principale faiblesse de TCPLearning reste son besoin en mémoire bien supérieur à TCP New Reno.

En 2017, Jiang *et al.* [Jia+17] proposent d'intégrer l'algorithme Q-learning dans TCP Vegas. Ils nomment leur proposition TCP-Gvegas. L'idée principale est d'utiliser Q-learning afin de déterminer la taille de la fenêtre de contention. Pour vérifier la pertinence de TCP-Gvegas, Jiang *et al.* ont implémenté leur algorithme sur le simulateur réseau *ns-2*. Ils l'ont ensuite comparé à TCP Vegas et TCP New Reno sur trois scénarios inclus avec *ns-2*. Ces scénarios ont la particularité d'utiliser des topologies sans-fil IEEE 802.11. D'après leurs simulations, TCP Gvegas permet de diminuer au moins de moitié le délai et d'augmenter d'un tiers le débit par rapport à TCP Vegas.

4.3.2 Application à la fonction de routage

Le routage fait partie des tâches les plus complexes des réseaux informatiques. Dans les années 90, deux tendances apparaissent afin de résoudre ce problème. Dans la première catégorie, il s'agit des algorithmes bio-inspirées comme les algorithmes utilisant les colonies de fourmis [Bon+98]. La deuxième approche s'inspire des algorithmes d'apprentissage par renforcement. Le premier de ces algorithmes est Q-routing [BL94]. Q-routing s'inspire de Q-learning. Nous verrons Q-routing plus en détails dans la section 4.4 qui lui est consacrée. Q-routing est le seul de ces algorithmes à avoir eu des travaux d'amélioration.

AdaR

AdaR [WW06] est un algorithme de routage conçu par Wang et Wang s'appuyant sur Q-learning. Sa particularité est d'utiliser l'ajustement des moindres carrés (*Least-Squares Policy Iteration*, LSPI). D'après leur évaluation sur une grille irrégulière de 400 capteurs, cette modification permet de diminuer le temps de convergence. Cela permet aussi d'obtenir de meilleurs résultats qu'avec Q-learning seul. D'après ses auteurs, AdaR a été conçu pour les réseaux de capteurs. Malheureusement, l'ajustement des moindres carrés augmente la complexité de l'algorithme. AdaR n'a pas été implémenté en tant que protocole de routage. Il a juste été évalué de manière analytique.

FROMS

FROMS [För+08] est un protocole de routage destiné aux réseaux de capteurs. Il a été conçu par Förster *et al.* et publié en 2008. Il utilise Q-learning afin de choisir la route. Dans leur article, ils comparent FROMS à Directed Diffusion, un protocole de routage pour les réseaux de capteurs [ST16].

D'après leur expérimentation, FROMS permet d'acheminer plus de paquets que Directed Diffusion. Son coût par paquets est aussi plus faible. Néanmoins, il est plus lent. Cependant, Directed Diffusion n'a pas été comparé à un protocole de routage standardisé. Il est donc difficile de juger de la pertinence de ces deux protocoles.

QELAR

QELAR [HF10] est un protocole de routage publié en 2010. Il est destiné aux réseaux de capteurs en environnements aquatiques. QELAR utilise Q-learning afin d'optimiser la durée de vie du réseau. Pour vérifier les performances de QELAR, Hu et Fei ont implémenté QELAR sur le simulateur *ns-2*. Ils l'ont ensuite comparé à VBF (*Vector-Based Forward*), un autre protocole de routage pour les réseaux de capteurs aquatiques. D'après leurs simulations, QELAR permet d'obtenir un taux de perte et un délai de livraison inférieur à VBF. De plus, QELAR améliore aussi la durée de vie du réseau.

4.4 Q-routing

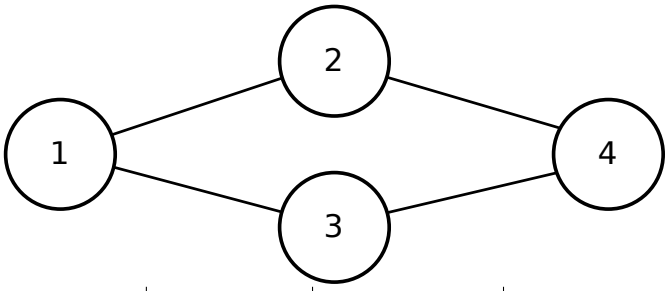
Dans cette section, nous verrons en détails Q-routing tel qu'il est décrit dans l'article de Boyan et Littman [BL94]. Nous avons choisi Q-routing car malgré sa date de publication, des travaux dérivés sont encore récemment publiés. Parmi les algorithmes spécialisés abordés à la section précédente, il est le seul avoir inspiré d'autres travaux.

4.4.1 Fonctionnement de Q-routing

Q-routing [BL94] est un algorithme de routage conçu par Boyan et Littman et publié en 1994. Il est inspiré de Q-learning [WD92], un algorithme d'apprentissage par renforcement publié deux ans plus tôt. Q-routing reprend certains concepts de Q-learning comme la Q-fonction, les Q-valeurs, mais surtout son système de mises à jour. Les Q-valeurs sont les métriques de routage de Q-routing. Elles sont calculées à partir de la Q-fonction qui associe une source x, une destination d et le prochain saut y tel que :

$$\Delta Q_x(d, y) = \eta(q + s + t - Q_x(d, y)) \tag{4.3}$$

où η est le taux d'apprentissage (généralement 0,5), q le nombre d'unités de temps passé en file d'attente, s le nombre d'unités de temps passé dans la transmission entre x et y, et t tel que : $t = \min_{z \in \text{voisins de } y} Q_y(d, z)$. Une route est donc définie à minima par son origine, sa destination, le prochain saut ainsi que la Q-valeur associée. Il peut donc y avoir plusieurs routes disponibles pour une même destination. La route effectivement utilisée est la route ayant la plus petite Q-valeur. La Figure 4.3 illustre les routes sur le nœud 1 et à destination du nœud 4. Sur cet exemple, la route vers le nœud 4 passant par le nœud 2 serait sélectionnée. Q-routing met à jour uniquement les Q-valeurs des routes sélectionnées. Sur l'exemple précédent, seule la route vers 4 passant par 2 aura sa Q-valeur actualisée. À la suite de l'actualisation d'une route, Q-routing choisit à nouveau la route avec la plus petite Q-valeur.



Source	Destination	Prochain saut	Q-valeur
1	2	2	10
1	3	3	10
1	4	2	20
1	4	3	20
1	2	3	30
1	3	2	30

FIG. 4.3 : Exemple de routes présentes sur le nœud 1 et à destination de 4 après exploration.

4.4.2 Analyse des résultats obtenus par Boyan et Littman

Dans leur article, Boyan et Littman ont évalué Q-routing sur plusieurs topologies réseaux dont leur grille irrégulière 6×6. La Figure 4.4 représente cette topologie réseau. Elle est composée de 36 nœuds répartis sur deux sous-grilles reliées par deux chemins de longueur différente. Boyan et Littman ont publié uniquement les résultats pour cette topologie particulière.

Q-routing est comparé à l'algorithme de Bellman-Ford, un algorithme de plus court chemin. D'après leur simulation, Bellman-Ford offre un meilleur délai de bout à bout à faible charge. En effet, à faible charge, Q-routing n'est pas toujours capable de trouver le chemin le plus rapide. À plus forte charge, Q-routing équilibre la charge entre deux chemins. Cela lui permet d'éviter une congestion sur le lien entre les nœuds 16 et 22. Q-routing arrive ainsi à maintenir un délai de livraison plus faible que Bellman-Ford. Cependant, lorsque les deux chemins reliant les deux sous-grilles sont saturés, le délai de livraison augmente fortement. Malheureusement, le taux de livraison des paquets n'est pas indiqué. Cette métrique est importante car Q-routing pourrait détruire des paquets afin que les paquets restants puissent être acheminés plus rapidement.

Les travaux de Boyan et Littman sont très difficilement reproductibles. Les simulations ont été effectuées sur leur simulateur développé en interne. Certaines simplifications sont présentes comme les files d'attente de taille illimitée. Les hypothèses de fonctionnement du simulateur ne sont pas toutes connues. Par exemple, bien que Q-routing soit distribuable, il n'est pas précisé s'il fonctionnait de façon distribuée sur leur simulateur. Chaque niveau de charge n'est pas défini. De plus, la répartition des couples sources-destinations n'est pas spécifiée. Sur une grille irrégulière comme celle-ci, cette répartition a une grande importance pour estimer les chemins utilisés. Enfin, Q-routing a

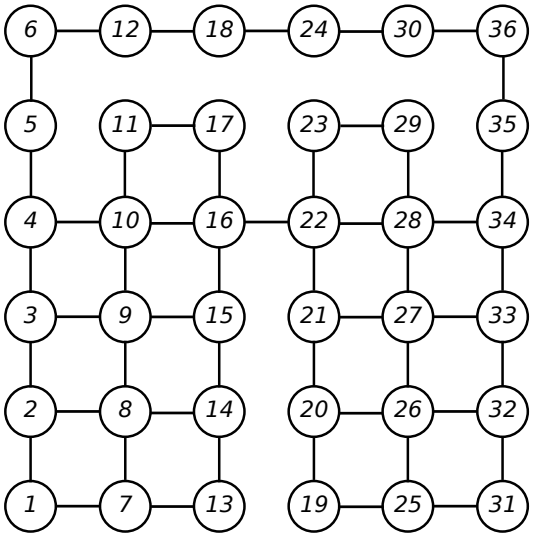


FIG. 4.4 : Grille irrégulière 6×6 de Boyan et Littman, extrait de [BL94].

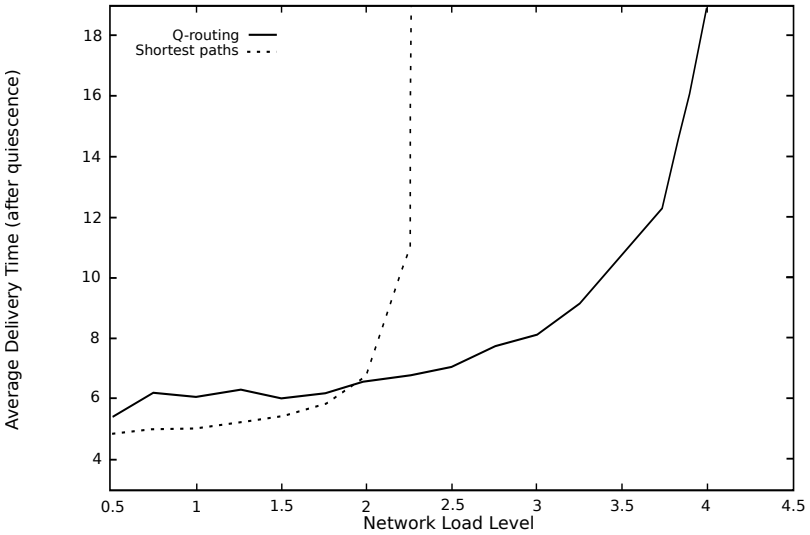


FIG. 4.5 : Délai de livraison en fonction de la charge réseau pour Q-routing et Bellman-Ford, extrait [BL94].

un comportement glouton à cause de sa stratégie de choix et de son système de mise à jour. Ce comportement se traduit par une sensibilité aux optimums locaux. Concrètement, une congestion même momentanée peut priver Q-routing de sa meilleure route.

4.5 Modifications génériques de Q-routing

Quelques années plus tard, plusieurs modifications ont été proposées pour combler les faiblesses de Q-routing. Predictive Q-routing est le premier d'entre eux. Son objectif est de corriger deux faiblesses. La première est le comportement glouton dû à la stratégie de Q-routing. La seconde faiblesse concerne les performances de Q-routing à faible charge.

4.5.1 Predictive Q-routing

En 1996, Choi et Yeung [CY96] proposent une amélioration de Q-routing. Ils l'appellent Predictive Q-routing (PQ-routing). PQ-routing a vocation de combler deux faiblesses. Premièrement, Q-routing a des difficultés à trouver le chemin optimal lorsque la charge réseau est faible. Choi et Yeung proposent d'utiliser un algorithme de plus court chemin lorsque la charge réseau est faible. Les effets de la stratégie gloutonne sont la seconde faiblesse que Choi et Yeung souhaitent atténuer. Pour cela, PQ-routing utilise 4 tables différentes : Q (comme Q-routing), B (meilleures Q-valeurs), R (taux de récupération des routes) et U (horodatage de la dernière mise à jour). Avec ces quatre tables, PQ-routing peut restaurer les anciennes Q-valeurs après une congestion. Contrairement à Q-routing, le taux d'apprentissage de PQ-routing est de 0,7. Choi et Yeung comparent PQ-routing à Q-routing sur deux topologies : une topologie originale de 15 nœuds et la grille irrégulière 6×6 de Boyan et Littman. L'évaluation porte uniquement sur le délai de bout-en-bout. PQ-routing offre un meilleur délai de bout-en-bout sur les différents scénarios incluant une variation du trafic réseau. Dans le cas où le trafic est important et apériodique, les résultats sont moins favorables pour PQ-routing. La durée d'apprentissage est aussi réduite grâce à PQ-routing. Cependant, comme Boyan et Littman, Choi et Yeung ont évalué leur algorithme sur leur simulateur développé en interne. Bien que certains paramètres de simulation soient donnés, l'emplacement des couples source-destination ne sont pas précisés pour la grille irrégulière.

4.5.2 Dual Reinforcement Q-Routing

Dual Reinforcement Q-routing (DRQ-routing) [Kum97] est un algorithme de routage dérivé de Q-routing. DRQ-routing a été conçu par Kumar et publié en 1997. Il apporte la rétropropagation des informations de la destination vers la source par rapport à Q-routing. Cette idée est issue de l'apprentissage par renforcement dual de Goetz *et al.* [GKM96]. Le taux d'apprentissage lors de la propagation est 0,7 et 0,9 lors de la rétropropagation. Le taux d'apprentissage de la rétropropagation est plus élevé, car la rétropropagation est considérée comme plus fiable que la propagation. Afin de valider la pertinence de la rétropropagation, Kumar compare DRQ-routing à Q-routing et l'algorithme de plus court chemin de Bellman-Ford. Il évalue ces trois algorithmes sur la grille irrégulière 6×6 de Boyan et Littman. L'évaluation porte uniquement sur le délai de livraison. Comme les travaux précédents, Kumar utilise leur simulateur développé en interne. Certaines hypothèses ne sont pas réalistes. Par exemple, la taille des files d'attente n'est pas limitée. D'après les simulations, DRQ-routing permet un apprentissage plus rapide à forte charge. DRQ-routing délivre les paquets

plus rapidement que Q-routing indépendamment de la charge réseau. Il reste un peu plus lent que Bellman-Ford à faible charge. Kumar estime que le surcoût lié à la rétropropagation est négligeable.

4.5.3 Q-Neural Routing

Q-Neural Routing est un algorithme dérivé de Q-routing conçu par Said Hoceini au cours de sa thèse de doctorat [Hoc04]. Sa particularité est d'utiliser un réseau de neurones à la place de la Q-fonction. Le réseau de neurones reçoit en entrée la destination et renvoie le prochain saut. D'après les simulations réalisées sur *OPNET*, Q-Neural Routing permet d'obtenir un délai de livraison plus court qu'avec Q-routing à moyenne et forte charge. Néanmoins, Q-Neural Routing a besoin de certaines hypothèses pour fonctionner. Par exemple, il est nécessaire de connaitre au préalable le nombre de destinations et de prochains sauts afin de pouvoir dimensionner le réseau de neurones. Cela implique aussi que la topologie soit statique. Le temps de convergence est aussi plus long avec Q-Neural Routing que Q-routing à cause de l'entraînement du réseau de neurones.

4.5.4 K-Shortest path Q-routing

K-Shortest path Q-routing est un algorithme hybride issue de la thèse de doctorat de Said Hoceini [Hoc04]. Il combine à la fois l'algorithme de Dijkstra généralisé et Q-routing. L'algorithme de Dijkstra permet de déterminer k plus courts chemins. Q-routing sélectionne un des k plus courts chemins. L'espace de recherche de K-Shortest path Q-routing est plus restreint que celui de la version originale de Q-routing car il se limite à k chemins. Il y a néanmoins au moins deux contreparties. D'après Hoceini, un mécanisme de détection des boucles de routage est nécessaire. De plus, il est nécessaire de pouvoir détecter les changements de topologies afin de recalculer les k plus courts chemins. Hoceini a implémenté Q-routing et son dérivé sur *OPNET*. Il a ensuite évalué K-Shortest path Q-routing sur deux topologies inspirées de la grille irrégulière de Boyan et Littman. D'après ses simulations, K-Shortest path Q-routing offre un meilleur délai de bout-en-bout que Q-routing et RIP, un protocole standardisé (RFC 1058) et implémentant l'algorithme de Bellman-Ford. Néanmoins, aucune autre métrique comme le taux de paquets livrés n'est fournie.

4.5.5 Q^2-routing

Q-routing ne traite pas les demandes particulières en termes de qualité de services. Par exemple, il n'est pas possible de paramétrer Q-routing pour optimiser la gigue. En 2018, Hendriks *et al.* [HCL18] proposent de remédier à ce besoin. Pour cela, ils créent Q^2-routing, un algorithme dérivé de Q-routing. Ils proposent d'ajouter trois coefficients C_d (délai), C_j (gigue) et C_l (taux de paquets livrés). Ces coefficients sont paramétrés en fonction des exigences en qualité de services. Il modifie la Q-fonction tel que :

$$Q_x(y,d) = (C_d \times C_l \times C_j) \times ((1-\alpha) \times Q_x(y,d) + \alpha(q+s+t)) \tag{4.4}$$

Hendriks *et al.* utilisent un taux d'apprentissage $\alpha = 0,5$. Ils proposent aussi l'utilisation d'une stratégie ε-glouton pour le choix des routes lors de la phase d'apprentissage. Enfin, Hendriks *et al.*

assurent que Q^2-routing est un protocole de routage hybride, complètement distribué et ne génère pas de boucle de routage. Afin de vérifier le fonctionnement et la pertinence de leurs modifications, Hendriks *et al.* ont implémenté Q^2-routing et Q-routing sur le simulateur réseau *ns-3*. Ils ont pu ainsi comparer leur implémentation de Q^2-routing à leur implémentation de Q-routing et AODV (RFC 3561), un protocole de routage ad-hoc. Ils proposent un scénario avec une topologie sans-fil avec trois chemins entre la source et la destination. Différentes perturbations apparaissent sur les trois chemins en cours de la simulation. Hendriks *et al.* utilisent le délai et le pourcentage de paquets conformes à une classe de trafic pour évaluer ces trois protocoles. D'après leurs simulations, Q^2-routing obtient dans l'ensemble de meilleurs résultats qu'AODV. Cependant, ces résultats ont été obtenus au prix d'un surcoût en termes de charge réseau.

4.6 Adaptation de Q-routing à des scénarios particuliers

Dans cette section, nous abordons trois adaptations de Q-routing pour trois scénarios particuliers : *i)* la mobilité, *ii)* les réseaux de capteurs et *iii)* la radio cognitive. Q-routing a été conçu pour les réseaux filaires. Ces scénarios n'ont donc pu être traités par Boyan et Littman.

4.6.1 Q-routing sur des scénarios avec mobilité

Q-routing a été créé avant IEEE 802.11. Il n'a pas été conçu pour les réseaux sans-fil et encore moins pour des scénarios avec mobilité. En 2020, Serhani *et al.* [SNJ20] entendent résoudre le cas de la mobilité avec Adaptive Q-routing (AQ-routing). AQ-routing est un algorithme dérivé de Q-routing. Il est spécialisé dans les scénarios avec mobilité. AQ-routing utilise une Q-fonction complexe s'appuyant sur la stabilité plutôt que la latence :

$$Q_{metric_{ij}} = \alpha_{ij} \cdot \varphi(MF_j) + (1 - \alpha_{ij}) \cdot \lambda ETX_{ij} \tag{4.5}$$

où i est un nœud, j un nœud voisin de i, MF est le facteur de mobilité, α_{ij} le taux d'apprentissage, γ un facteur de normalisation pour la fonction ETX et $\varphi(MF_j)$ est défini tel que :

$$\varphi(MF_j) = \frac{a}{1 - e^{\frac{-MF_j}{b}}} \tag{4.6}$$

Ils utilisent un taux d'apprentissage $\alpha = 0,95$ pour leurs simulations. Dans leur article, ils comparent AQ-routing à OLSR (*Optimized Link State Routing*, standard et ETX) sur le simulateur *ns-3*. Afin d'obtenir des performances optimales, leur implémentation d'AQ-routing s'inspire de certains concepts d'OLSR, un protocole de routage standardisé et spécialisé pour les réseaux MANETs (*Mobile Ad-hoc NETworks*). Par exemple, leur implémentation reprend les paquets HELLO ainsi que les paquets TC (Topology Control). Serhani *et al.* évaluent AQ-routing sur un cas statique et un cas avec mobilité. Le taux de paquets livrés et le délai de livraison sont les métriques d'évaluation. La topologie est composée de 30 nœuds répartis aléatoirement sur espace de 1000×1000 m. Dans le cas

statique, AQ-routing offre un taux de paquets livrés quasi-identique à OLSR-ETX. Cependant, AQ-routing achemine en moyenne les paquets plus lentement que les deux variants d'OLSR, soit 205 ms contre 173 ms pour OLSR-ETX. À basse vitesse (entre 0 et 3 m/s), OLSR-ETX offre les meilleures performances. Le taux de paquets livrés avec OLSR-ETX est équivalent à celui d'AQ-routing. OLSR-ETX livre aussi les paquets plus rapidement. À partir d'une mobilité 3 m/s, les résultats sont favorables à AQ-routing que ce soient en termes de taux de paquets livrés qu'en délai de livraison. Cependant, Serhani *et al.* ont augmenté la complexité de la Q-fonction utilisant la métrique ETX ainsi que d'autres coefficients pour obtenir ces résultats.

4.6.2 Q-routing pour les réseaux de capteurs

Q-routing n'est pas conçu spécifiquement pour les réseaux sans-fil de capteurs (*Wireless Sensor Network*, WSN). Bouzid *et al.* [Bou+20] proposent R2LTO (*Reinforcement Learning for LT Optimisation*), une version de Q-routing adaptée aux réseaux de capteurs. Leur changement porte essentiellement sur la récompense. Ils utilisent l'énergie pondérée par le nombre de sauts. Ils ont comparé leur algorithme à Q-routing et RLBR, un autre algorithme spécialisé, sur leur simulateur. D'après leur simulation, R2LTO améliore la durée de vie du réseau jusqu'à 25 %. Les autres métriques de qualité de service ne sont pas présentées. De plus, les simulations portent uniquement sur l'algorithme et non sur une éventuelle implémentation protocolaire. Le surcoût lié à la gestion du routage n'est donc pas évalué.

4.6.3 Q-routing pour la radio cognitive

La radio cognitive s'appuie sur la radio logicielle soit pour optimiser les couches physiques, voire liaisons des réseaux sans-fil, soit pour réutiliser certaines bandes de fréquences sous-utilisées hors ISM. Dans le deuxième cas, le routage peut s'avérer délicat car le réseau peut changer fréquemment de bandes de fréquences ou être sur plusieurs bandes de fréquences. Xia *et al.* [Xia+09] proposent d'adapter Q-routing et DRQ-routing pour la radio cognitive. Afin d'évaluer les performances de leurs modifications, ils les ont implémentés sur le simulateur *OMNet++*. Ils sont comparés à un protocole du plus court chemin optimisé pour la radio cognitive. D'après leurs simulations, Q-routing et DRQ-routing offrent en moyenne un délai de livraison plus court que le protocole type plus court chemin. DRQ-routing offre en moyenne le meilleur délai de livraison. Son temps de convergence est aussi plus court de que celui de Q-routing.

4.7 Conclusions

Dans ce chapitre, nous avons abordés quelques algorithmes d'apprentissage par renforcement. Parmi eux, Q-learning est un algorithme qui offre un bon rapport entre efficacité et simplicité de mise en œuvre. Par exemple, trois des quatre algorithmes de routages présentés utilisent Q-learning ou s'en inspirent. En effet, Q-learning ne nécessite ni modèle, ni jeu de données initial. Parmi tous les travaux cités, nous nous sommes concentrés sur l'algorithme de routage Q-routing. Q-routing offre

des résultats prometteurs sur le simulateur de Boyan et Littman [BL94]. Au fil des années, de nombreuses modifications sont apparues comme Predictive Q-routing [CY96] ou Q^2-routing [HCL18]. Nous avons vu que certaines modifient les fonctionnements de Q-routing comme Q-Neural Routing [Hoc04]. D'autres modifications sont destinées à adapter Q-routing à une situation particulière comme la radio cognitive [Xia+09]. Dans la prochaine partie, nous aborderons nos contributions : l'implémentation et les améliorations Q-routing.

Conclusion de la première partie

Dans cette partie, nous avons présenté un état de l'art en lien avec le sujet. Dans le Chapitre 1, nous avons commencé par présenter l'Industrie 4.0. Contrairement aux précédentes révolutions industrielles, l'objectif n'est pas d'augmenter la capacité de production, mais d'améliorer la flexibilité et la gestion des ressources. Pour cela, l'Industrie 4.0 fixe des objectifs techniques et organisationnels afin d'accroître la compétitivité de l'industrie [Kag13]. D'après les objectifs que nous avons détaillés, l'Industrie 4.0 est plus une évolution ou une formalisation des pratiques existantes qu'un changement radical. La particularité de l'Industrie 4.0 est qu'elle repose sur l'Internet des Objets et les systèmes cyber-physiques, ce qui entraîne quelques contraintes supplémentaires notamment pour les réseaux informatiques.

Dans le Chapitre 2, nous avons abordé ensuite les technologies des réseaux industriels. Une partie des technologies industrielles sont apparues parce que les technologies des réseaux informatiques n'étaient pas adaptées. Cela a contribué à la création d'une quinzaine de bus de terrain [Dec09]. À partir des années 2000, des technologies dérivées de standards IEEE sont apparues. C'est le cas de SERCOS III, POWERLINK et EtherCAT comme dérivées d'Ethernet, et WIA-PA, ISA100.11a et WirelessHART pour IEEE 802.15.4. À partir de 2010, Ethernet et IEEE 802.11 ont évolué pour prendre en compte les contraintes industrielles comme le temps réel. Cela a donné naissance à Ethernet TSN (*Time-Sensitive Networking*) et IEEE 802.15.4 TSCH (*Time Slotted Channel Hopping*). Ce sont deux exemples de la convergence entre les réseaux informatiques et les réseaux industriels.

Dans le Chapitre 3, nous avons présenté trois axes afin d'améliorer les performances d'un réseau informatique : *i)* l'atténuation des interférences, *ii)* l'optimisation inter-couches et *iii)* les réseaux cognitifs. La plupart des technologies des réseaux sans-fil utilisent les mêmes bandes de fréquences, ce qui favorise l'apparition d'interférences. Ces interférences entraînent la destruction des trames et donc des données. Par exemple, lorsque IEEE 802.11 et IEEE 802.15.4 entrent en concurrence, jusqu'à 80 % des trames IEEE 802.15.4 peuvent être détruites [Pet+07].

Les optimisations inter-couches reposent sur la communication entre des couches protocolaires potentiellement non adjacentes. Bien que des travaux comme [Car+19] montrent des résultats encourageants, les optimisations inter-couches sont très difficilement pérennes. Elles sont généralement abandonnées comme nous l'avons montré avec les *drafts* IETF (Section 3.3.3).

Les réseaux cognitifs sont une généralisation de la radio cognitive à l'ensemble des couches du modèle OSI. Afin d'atteindre un ou plusieurs objectifs de performances, ils peuvent reconfigu-

rer automatiquement les différents paramètres des couches protocolaires via des optimisations inter-couches par exemple. Ces objectifs de performances peuvent être la maximisation du débit, la minimisation de la latence ou l'amélioration de la fiabilité. Néanmoins, en plus des problèmes de pérennité, les réseaux cognitifs sont si complexes que la majorité des travaux se sont arrêtés avant la mise en œuvre. Face à cette complexité, la plupart des auteurs ont préféré se concentrer sur l'amélioration d'une couche protocolaire ou d'une fonctionnalité.

Dans le Chapitre 4, nous nous sommes focalisés sur l'apprentissage par renforcement et ses applications dans les réseaux informatiques. L'apprentissage par renforcement est l'un des trois paradigmes de l'apprentissage automatique. Sa particularité est d'apprendre par essais et erreurs. De plus, certains algorithmes comme Q-learning ne nécessitent pas de jeu de données initial. De tous les exemples d'applications présentés, Q-routing est un des rares à avoir inspiré d'autres travaux. Q-routing est un algorithme de routage inspiré de Q-learning, un algorithme d'apprentissage par renforcement. Certaines propositions améliorent le fonctionnement de Q-routing comme PQ-routing [CY96]. D'autres améliorations permettent de l'adapter à un cas d'utilisation précis comme la radio cognitive [Xia+09]. Les résultats obtenus sont intéressants, mais restent difficilement reproductibles dans des environnements réalistes. Dans la prochaine partie, nous proposons une implémentation de Q-routing et son évaluation. Nous proposons aussi des améliorations comme l'atténuation des effets de la stratégie gloutonne.

partie II

Contributions

Introduction de la deuxième partie

Dans la première partie, nous avons abordé l'Industrie 4.0 et ses objectifs. Nous avons ensuite détaillé quelques technologies compatibles avec les réseaux de l'Industrie 4.0. Nous avons poursuivi avec trois axes d'amélioration pour les réseaux notamment sans-fil : *i)* l'atténuation des interférences, *ii)* l'optimisation inter-couches et les réseaux cognitifs. Ces derniers sont si complexes que nous avons préféré nous concentrer sur l'amélioration d'une fonctionnalité. Nous nous sommes alors concentrés sur l'apprentissage par renforcement et son application pour les réseaux informatiques. Parmi toutes les applications, nous avons sélectionné Q-routing [BL94], un algorithme de routage aux résultats prometteurs. Néanmoins, ces résultats ont été obtenus sur leur simulateur développé en interne. De plus, le délai de bout-en-bout est la seule métrique présentée. Les résultats de Q-routing ont donc besoin d'être confirmés dans un environnement plus réaliste. Dans cette seconde partie, nous allons détailler nos contributions : l'implémentation et l'amélioration de Q-routing ainsi que leurs évaluations. Dans le Chapitre 5, nous détaillons les caractéristiques de notre implémentation ainsi que les choix que nous avons fait (première contribution). Nous proposons une première évaluation de notre implémentation sur deux topologies filaires dont la grille irrégulière de Boyan et Littman [BL94] (seconde contribution). Nous utilisons OLSRv2 (RFC 7181) comme protocole de routage de référence.

Q-routing utilise une stratégie gloutonne. Elle simplifie sa mise en œuvre mais rend Q-routing sensible aux optimums locaux. Boyan et Littman reconnaissent qu'une congestion momentanée peut priver Q-routing d'une route contrairement à un algorithme utilisant le plus court chemin. Dans le Chapitre 6, nous nous concentrons sur cet effet de la stratégie gloutonne. Nous mettons en évidence ce phénomène et nous proposons deux atténuations (troisième contribution) : *i)* les mises à jour ε-glouton et *ii)* le système d'époques. Enfin, nous proposons une première évaluation de notre implémentation sur deux topologies sans-fil incluant une version adaptée de la grille irrégulière de Boyan et Littman (quatrième contribution).

Dans le Chapitre 7, après avoir mis en évidence des problèmes d'estimation sur la métrique de routage, nous proposons d'utiliser le filtrage pour résoudre ce problème (cinquième contribution). Nous évaluons notre proposition sur deux scénarios. Enfin, nous proposons une première évaluation de notre implémentation sur un scénario avec mobilité (sixième contribution).

Chapitre 5

Q-routing : de l'algorithme au protocole

Sommaire

6.1 Introduction **87**
6.2 Mise en évidence des effets de la stratégie gloutonne **88**
6.3 Solutions proposées **90**
6.3.1 Approche probabiliste inspirée de l'apprentissage par renforcement . . 90
6.3.2 Système d'époque 91
6.4 Évaluation des solutions proposées **92**
6.4.1 Exemple jouet sans-fil 92
6.4.2 Grille sans-fil inspirée de Boyan et Littman 96
6.5 Conclusions **103**

5.1 Introduction

Q-routing [BL94] est un algorithme de routage inspiré de Q-learning [WD92], un algorithme d'apprentissage par renforcement. Comme nous l'avons vu dans la Section 4.4, Boyan et Littman ont évalué Q-routing en tant qu'algorithme dans leurs travaux. C'est-à-dire sans prendre en compte, par exemple, les échanges nécessaires au bon fonctionnement du protocole en mode distribué. De plus, Q-routing a été évalué sur leur simulateur développé en interne, ce qui rend leurs résultats difficilement reproductibles. Surtout, les hypothèses de fonctionnement ne sont pas connues. Enfin, l'algorithme est très succinctement décrit. Bien que Q-routing ait été implémenté sur OPNET et ns-3 notamment par Hoceini [Hoc04] et Hendriks *et al.* [HCL18], il n'y a pas d'implémentation publiquement disponible. Dans ce chapitre, nous proposons une implémentation de Q-routing sur le simulateur réseau Qualnet. Nous évaluons notre implémentation sur un premier scénario afin de vérifier sa capacité à utiliser un chemin alternatif, c'est-à-dire la principale caractéristique de

Q-routing. Dans un second temps, nous évaluons Q-routing dans un scénario utilisant la grille irrégulière de Boyan et Littman. Nous choisissons OLSRv2 (RFC 7181), un protocole de routage standardisé comme protocole de routage de comparaison. Nous montrons que notre implémentation de Q-routing est capable d'utiliser un chemin alternatif contrairement à OLSRv2. Cela permet à Q-routing de livrer plus de paquets et plus rapidement que OLSRv2 sur les scénarios proposés. Dans la Section 5.2, nous commencerons avec l'implémentation de Q-routing sur Qualnet. Nous verrons notamment les différents détails d'implémentation ainsi que les hypothèses de fonctionnement. Dans la Section 5.3, nous évaluerons notre implémentation sur deux scénarios et grâce à trois métriques (taux de paquets reçus, délai de livraison, variation de délai). Nous terminerons par une conclusion dans la Section 5.4.

5.2 Implémentations de Q-routing

Nous allons maintenant détailler notre implémentation de Q-routing. Elle a été réalisée sur le simulateur réseau à événements discrets Qualnet [SVS20]. Qualnet est un simulateur réseau commercial dérivé de GloMoSim. Il a l'avantage de supporter aussi bien certaines distributions GNU/Linux que Microsoft Windows. De plus, il supporte bien la mise à l'échelle, notamment via des calculs multi-cœurs. Enfin, Qualnet est écrit en C++ et C. Il est donc possible de développer notre protocole dans l'un de ces langages. Notre implémentation est inspirée d'une implémentation basique de l'algorithme du plus court chemin Bellman-Ford fournie avec Qualnet. Cette implémentation fournit des fonctionnalités essentielles comme la vérification de la péremption des routes et un système de mises à jour. Notre implémentation de Q-routing est proactive et totalement distribuée. Chaque nœud n'utilise que des informations obtenues localement. La table de routage est donc construite à partir des informations échangées entre les nœuds. La complexité d'implémentation de Q-routing réside surtout dans la mesure et la collecte du délai nécessaire au calcul de la Q-valeur.

5.2.1 Mesure du délai et format des entêtes

Q-routing utilise la durée en file d'attente et la durée de transmission jusqu'au prochain voisin pour calculer la Q-valeur. Ces deux valeurs forment finalement le délai. Il existe plusieurs moyens de mesurer le délai comme avec des paquets ICMP Echo/Reply, des paquets spécialisés ou des paquets de routages.

Les paquets ICMP Echo permettent d'obtenir une mesure du délai aller-retour indépendamment de la synchronisation de l'horloge des nœuds. Cette méthode est moins intéressante sur les réseaux où l'encombrement du lien est dépendant du sens. En effet, le lien peut être encombré uniquement dans un sens. Le résultat de la mesure est la durée de l'aller-retour, il ne sera pas possible de distinguer quel sens est congestionné. Qualnet n'implémente pas les paquets ICMP Echo Request. Il n'est donc pas possible d'utiliser cette technique avec Qualnet. Il reste néanmoins possible de réimplémenter cette méthode en utilisant les paquets du protocole de routage.

Dans leur article, Boyan et Littman proposent pour chaque paquet de renvoyer un paquet de la destination vers la source contenant notamment le délai de bout-en-bout. Cette méthode a au moins deux inconvénients. Premièrement, afin d'obtenir le délai de bout-en-bout, il faut que les paquets soient horodatés et que les horloges des nœuds soient synchronisées. Le second inconvénient est le trafic de données généré. Q-routing a été conçu à l'origine pour les réseaux filaires. Sur des liens où le flux montant n'impacte pas le flux descendant, les deux trafics peuvent être présents sans se perturber. Sur les réseaux sans-fil, ce n'est pas le cas. Ce flux de données « retour » peut détériorer les performances en occupant une part du média. Une solution intermédiaire serait de diminuer la fréquence d'envoi du paquet de retour.

Nous avons préféré mesurer le délai grâce aux paquets d'échanges de routes. Les paquets d'échange de routes sont des paquets de type UDP. L'entête des paquets comprend le nombre de routes et un horodatage. Cet horodatage est ajouté juste avant la mise en file d'attente du paquet de routage. Le délai est calculé par le nœud réceptionnant le paquet en comparant cet horodatage et sa propre horloge. Le délai mesuré correspond la durée passée en file d'attente et la durée de la transmission entre la source et son voisin direct, soit les termes $q + s$ de la Q-fonction. La mesure du délai est ainsi renvoyée lors du prochain envoi de mise à jour. Cette méthode permet de diminuer le nombre de paquets liés au routage. Néanmoins, elle a plusieurs inconvénients. En effet, pour fonctionner, il est nécessaire que les horloges des nœuds soient synchronisées. La précision de la mesure du délai dépend donc de la précision de la synchronisation. Autre inconvénient, la gestion des files d'attente peut fausser la mesure de la durée moyenne passée en file d'attente. Pour illustrer cette situation, prenons par exemple deux nœuds avec chacun au moins deux files d'attente. La politique de ces files consiste à envoyer les paquets en fonction de leur priorité. Les paquets de routage mesureront le délai correspondant aux paquets de leur priorité et non le délai moyen.

5.2.2 Cycle de vie des routes

Une route est définie par le triplet source, destination, prochain saut. En complément, il y a un numéro de séquence, la Q-valeur, le nombre de sauts, un horodatage local ainsi que le statut de la route pour indiquer si la route est active. En effet, si la route est active, sa Q-valeur sera mise à jour. Le numéro de séquence est défini à partir de l'horloge du nœud créant la route et n'est plus modifié ensuite. La Q-valeur est initialisée à 0. Il s'agit d'une initialisation optimiste car elle favorise l'exploration des routes. L'horodatage est local et indique la dernière mise à jour de la route.

Notre implémentation de Q-routing est proactive. Dans l'algorithme original, les routes actives sont mises à jour à chaque paquet de données. Afin d'économiser de la bande passante, nous avons choisi d'utiliser des mises à jour indépendantes des paquets de données. Toutes les 10 s, chaque nœud envoie toutes les routes valides de sa table de routage à ses voisins immédiats. Jusqu'à 32 routes sont envoyées par paquet. La diffusion de nouvelles routes via les mises à jour périodiques serait lente. En effet, il faudrait, dans le pire des cas, 10 s par nœud intermédiaire. En complément, lorsqu'une nouvelle route est reçue, une mise à jour spontanée est directement envoyée afin de diffuser cette route plus rapidement.

Si la route contenue dans la mise à jour fait plus de 16 sauts, la route est rejetée. Il s'agit d'un

paramètre provenant de l'implémentation basique de Bellman-Ford que nous avons conservé. La route est mise à jour si l'horodatage de la mise à jour est plus récent que celui de la route présente localement. Si les horodatages sont identiques, le deuxième critère est la Q-valeur sur les sauts précédents. Si la Q-valeur de la mise à jour est plus petite que celle des précédents sauts de la route locale, la mise à jour est acceptée. Si la route mise à jour est une route active, alors sa Q-valeur est mise à jour. Si une route ne reçoit aucune mise à jour pendant 60 s, elle est considérée comme périmée. La Q-valeur est mise à jour avec une pénalité. Cette pénalité favorise les routes qui n'ont jamais été périmées.

Le choix des routes actives se produit dans deux cas : *i)* après la mise à jour de la Q-valeur d'une route active ; *ii)* à la création d'une nouvelle route. Q-routing choisit par défaut toujours la route valide ayant la plus petite Q-valeur. Si plusieurs routes valides ont la Q-valeur minimale, alors Q-routing choisit une route aléatoirement parmi ces routes. Ces paramètres liés au cycle de vie des routes sont modifiables via un fichier de configuration afin de les adapter à un scénario ou une topologie particulière. En effet, un scénario incluant de la mobilité nécessite plus de mises à jour et une péremption plus rapide des routes qu'un scénario statique.

5.2.3 Choix du taux d'apprentissage η

Le taux d'apprentissage η permet de déterminer la prépondérance des mises à jour par rapport à la valeur initiale. Plus le paramètre η est proche de 1, plus une valeur récente aura de l'importance dans le calcul de la Q-valeur. Cela impacte directement la réactivité de Q-routing aux changements, mais aussi sa sensibilité aux erreurs de mesures et événements éphémères comme les congestions momentanées. Boyan et Littman [BL94] proposent initialement d'utiliser un taux d'apprentissage de 0,5. Kumar [Kum97] propose d'utiliser un taux d'apprentissage de 0,7. Choi et Yeung [CY96] utilisent quant à eux un taux de 0,9. Serhani *et al.* [SNJ20] proposent d'utiliser un taux de 0,95. Il n'y a pas de valeur faisant autorité pour le taux d'apprentissage. D'après nos expérimentations sur notre exemple jouet sans-fil (voir chapitre suivant, Section 6.4.1), un taux d'apprentissage de 0,9 permet d'obtenir les meilleurs résultats en termes de taux de paquets livrés et de délai de livraison.

5.3 Évaluation de l'implémentation

Nous allons évaluer notre implémentation sur plusieurs scénarios : un cas d'exemple et la grille irrégulière de Boyan et Littman. L'ensemble des résultats ont été obtenus avec Qualnet 9.2.1.

5.3.1 Exemple jouet

Avant d'aborder un scénario complexe, nous allons commencer par un scénario relativement simple. La topologie est représentée par la Figure 5.1. Nous utilisons des liens filaires IEEE 802.3 de seulement 10 Mb/s afin qu'ils soient saturés plus rapidement. La simulation dure 15 minutes. Le nœud 1 envoie des données applicatives de 512 octets vers le nœud 4 à partir de la première seconde jusqu'à l'avant-dernière minute. Le débit est d'un paquet toutes les 10 ms soit 410 kb/s.

Un trafic d'arrière-plan apparait au bout de 8 minutes entre les nœuds 2 et 3. Il ne s'arrête qu'à l'avant-dernière minute. Le débit du trafic d'arrière-plan est constant au cours d'une simulation. La Figure 5.1 précise aussi l'emplacement des flux de données. Pour évaluer Q-routing, nous utilisons l'implémentation d'OLSRv2 de l'Université de Niigata (nuOLSRv2) comme référence. Le Tableau 5.1 récapitule l'ensemble des paramètres de simulation. L'objectif de ce scénario est de montrer la capacité de Q-routing à utiliser un chemin alternatif dans une topologie filaire simple.

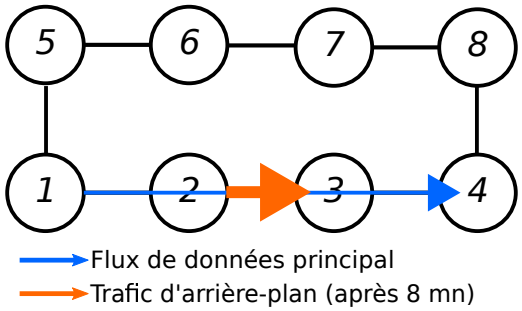


FIG. 5.1 : Topologie de notre exemple jouet et placement des flux de données. Les chiffres sont les identifiants des nœuds.

Élément	Paramètre	Valeur
Simulation	Durée simulée	15 min.
	Graines	30 graines différentes
Réseau	Lien	Lien filaire IEEE 802.3 10 Mb/s
	Temps de propagation	1 ms
Noeud	File d'attente	1 file FIFO
Données applicatives	Taille par envoi	512 octets
	Début	1 s
	Fin	14 min.
Flux d'arrière-plan	Taille par envoi	512 octets
	Début	8 min.
	Fin	14 min.
Q-routing	η	0,9
	Phase de découverte	15 s

TAB. 5.1 : Paramètres de simulation

Résultats

Pour évaluer les performances des deux protocoles de routage, nous avons relevé le taux de paquets livrés, le délai de bout-en-bout et la gigue (RFC 3393) du flux de données applicatives. Le

débit du flux d'arrière-plan est exprimé au niveau applicatif. Les paquets désordonnés sont détruits par la destination.

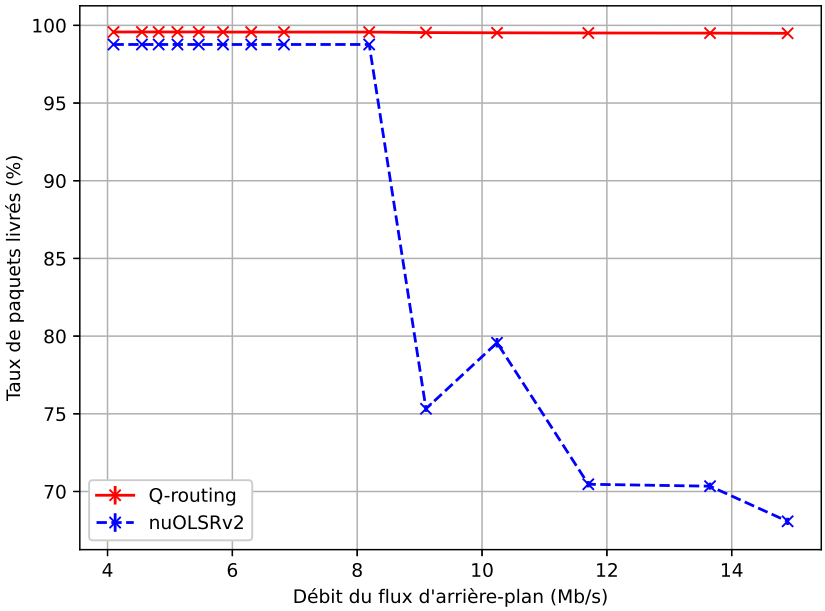


FIG. 5.2 : Taux de paquets livrés en fonction du débit d'arrière-plan avec notre scénario jouet.

La Figure 5.2 illustre le taux de paquets livrés en fonction du débit du flux d'arrière-plan pour Q-routing et nuOLSRv2. Entre 4 et 8 Mb/s, le taux de livraison de Q-routing est légèrement plus élevé que celui nuOLSRv2. Comme la source de trafic principal commence à émettre à la première seconde, les tables de routage ne sont pas encore constituées. Cette différence de taux de paquets reçus mesure indirectement la rapidité de propagation des routes. Contrairement à notre implémentation de Q-routing, nuOLSRv2 ne dispose que de mises à jour périodiques. Ceci ralentit la propagation des routes car il doit attendre la découverte des nœuds (toutes les 2 s) puis la propagation des informations de topologie (toutes les 5 s). Les paquets arrivés dans le désordre sont détruits par la destination. D'après le taux de paquets livrés de nuOLSRv2, le lien entre les nœuds 2 et 3 sature entre 8 et 9 Mb/s. Le taux de paquets livrés diminue fortement sur cet intervalle. Contrairement à nuOLSRv2, Q-routing utilise le chemin alternatif passant par les nœuds 5, 6, 7 et 8. Les paquets ne sont donc pas détruits à cause de la congestion.

La Figure 5.3 illustre le délai moyen de livraison en fonction du débit du flux d'arrière-plan pour Q-routing et nuOLSRv2. Le délai de livraison est identique pour les deux protocoles entre 4 et 8 Mb/s. Suite à la saturation du lien, le délai de livraison augmente fortement avec nuOLSRv2. À partir de 10 Mb/s, le délai de livraison diminue car le taux de paquets livrés diminue aussi. En effet, comme une partie des paquets est détruite, les paquets restants rencontrent moins de congestion jusqu'à la destination. Il augmente très légèrement avec Q-routing car le chemin alternatif est plus long. Il

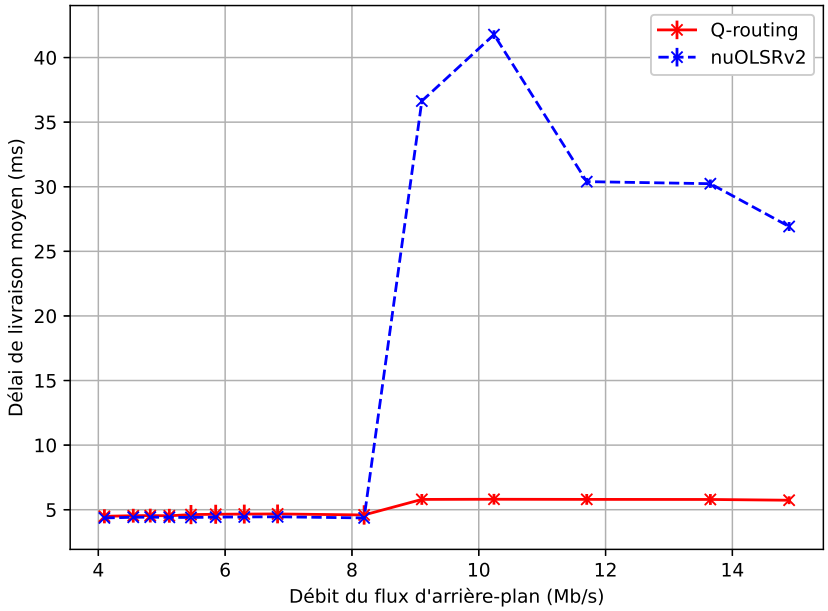


FIG. 5.3 : Délai de livraison moyen en fonction du débit d'arrière-plan avec notre scénario jouet.

y a un point singulier en termes de taux de paquets délivrés pour nuOLSRv2 autour de 10 Mb/s. Il correspond au pic en termes de délai de livraison.

La Figure 5.4 illustre la gigue moyenne en fonction du débit du flux d'arrière-plan pour Q-routing et nuOLSRv2. Les deux courbes ont le même profil mais la gigue est en moyenne plus élevée avec Q-routing qu'avec nuOLSRv2, indépendamment du débit. Les variations de délais sont plus importantes avec Q-routing qu'avec nuOLSRv2 même si le délai de livraison moyen est plus faible.

5.3.2 La grille irrégulière de Boyan et Littman

Nous avons vu que Q-routing offre de bons résultats dans un scénario qui lui est favorable. En effet, d'après les résultats de Boyan et Littman, Q-routing est capable d'utiliser un chemin alternatif pour répartir la charge sur différents chemins. Nous allons maintenant évaluer Q-routing sur la grille irrégulière de Boyan et Littman. La topologie est représentée par la Figure 5.5. Cette topologie est constituée de 36 nœuds reliés entre eux par des liens filaires 10 Mb/s. L'objectif est d'observer le comportement de Q-routing à forte charge réseau sur la grille irrégulière de Boyan et Littman et en présence de trafic alterné. D'après l'article de Boyan et Littman [BL94], Q-routing obtient de bons résultats y compris avec un trafic alterné. Pour évaluer Q-routing, nous utilisons l'implémentation d'OLSRv2 de l'Université de Niigata (nuOLSRv2) comme référence. La simulation dure 61 minutes.

Il y a 8 flux de données. 4 flux sont disposés horizontalement (Figure 5.6) et les 4 autres verticalement (Figure 5.7). Le chronogramme Figure 5.8 indique leur emplacement et leur période d'acti-

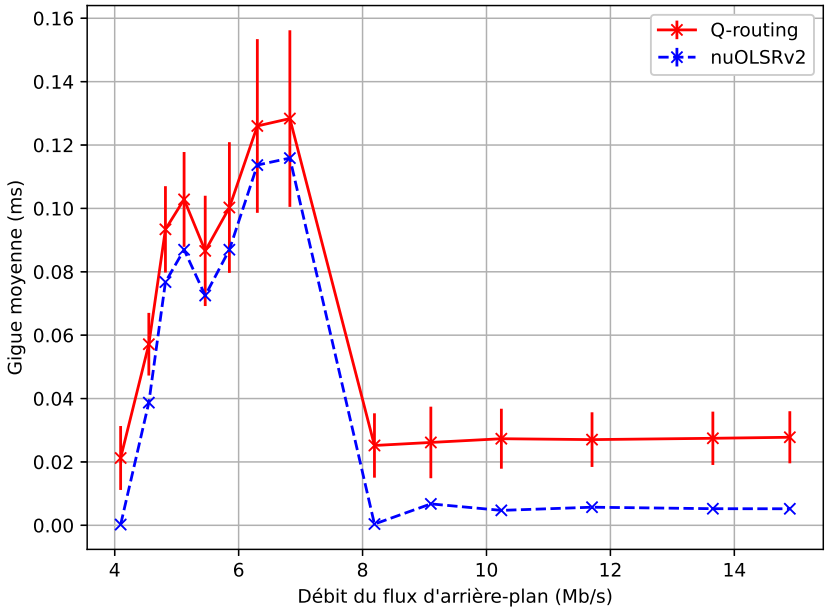


FIG. 5.4 : Gigue moyenne en fonction du débit d'arrière-plan avec notre scénario jouet.

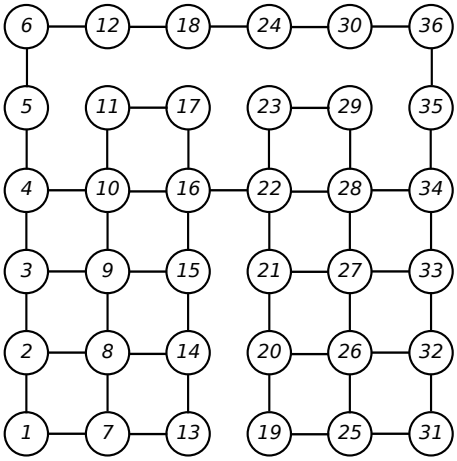


FIG. 5.5 : La grille irrégulière 6×6 de Boyan and Littman [BL94]

vité. Cette période d'activité dure 9 minutes et est suivie d'une minute d'absence d'envoi. Le trafic est donc alterné spatialement et temporellement. Afin que les routes soient établies, les sources commencent leur envoi une minute après le début de la simulation. Les flux de données sont synchronisés et ont le même débit. Les données applicatives ont une taille de 512 octets par envoi. Les paquets reçus dans le désordre sont détruits par la destination. Le Tableau 5.2 récapitule l'ensemble

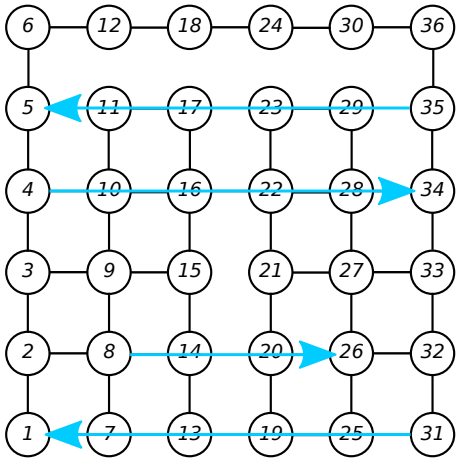


FIG. 5.6 : Placement des 4 flux horizontaux.

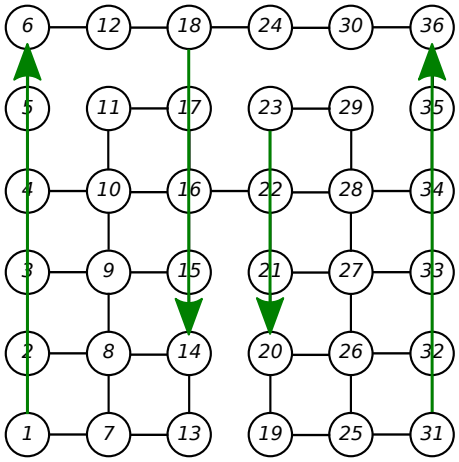


FIG. 5.7 : Placement des 4 flux verticaux.

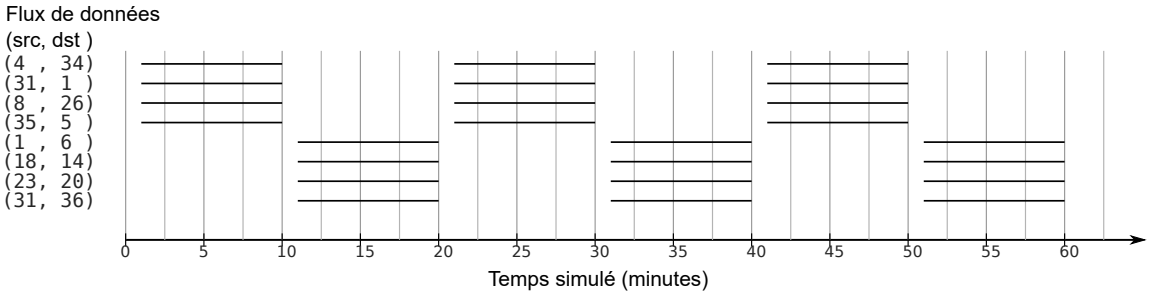


FIG. 5.8 : Chronogramme des flux de données sur la grille irrégulière.

Element	Paramètre	Valeur
Simulation	Durée simulée	61 min.
	Graines	30 graines différentes
Réseau	Lien	Lien filaire 10 Mb/s
	Temps de propagation	1 ms
Noeud	File d'attente	1 file FIFO
Données applicatives	Taille par envoi	512 octets
Q-routing	η	0,9
	Phase de découverte	15 s

TAB. 5.2 : Paramètres de simulation

des paramètres de simulation.

Résultats

Pour évaluer les performances des deux protocoles de routage, nous avons relevé le taux moyen de paquets livrés, le délai moyen de bout-en-bout et la gigue des flux de données applicatives. Le débit est exprimé au niveau applicatif et par flux de données.

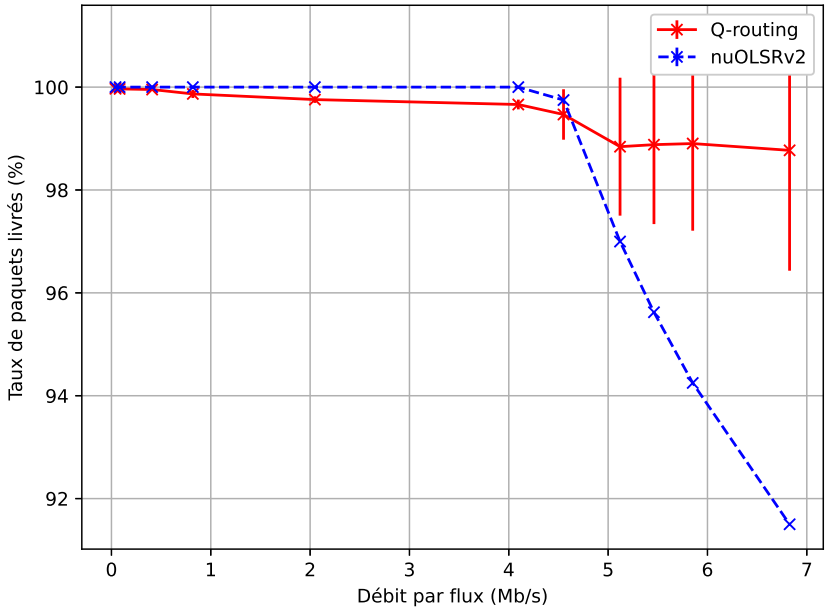


FIG. 5.9 : Taux de paquets reçus pour l'ensemble des sources en fonction du débit sur la grille irrégulière.

La Figure 5.9 illustre le taux de paquets reçus en fonction du débit par flux. nuOLSRv2 livre tous les paquets jusqu'à 4 Mb/s. À partir de 4,5 Mb/s, le taux de paquets livrés chute rapidement. Q-routing livre moins de paquets jusqu'à 4,5 Mb/s. Le taux de paquets livrés reste supérieur à 99 %. À partir de 4,5 Mb/s, Q-routing livre en moyenne plus de paquets que nuOLSRv2. Néanmoins, l'écart-type est élevé.

La Figure 5.10 illustre le délai de livraison moyen en fonction du débit par flux. Jusqu'à 4 Mb/s, nuOLSRv2 est légèrement plus rapide que Q-routing. La différence reste inférieure à 1 ms. À partir de 4,5 Mb/s, Q-routing offre en moyenne un meilleur délai de livraison que nuOLSRv2. À forte charge, l'écart-type du délai de livraison est élevé que soit pour Q-routing ou nuOLSRv2.

La Figure 5.11 représente la gigue moyenne en fonction du débit su la grille irrégulière. nuOLSRv2 offre la meilleure gigue indépendamment du débit. La gigue augmente lentement entre 0 et 4,5 Mb/s. Entre 4,5 Mb/s et 5,5 Mb/s, la gigue moyenne est multipliée par 5 par nuOLSRv2. Pour Q-routing, la gigue moyenne se situe entre 0,08 ms et 0,10 ms. La gigue moyenne est importante mais son amplitude de variation est plus fiable qu'avec nuOLSRv2. Contrairement au scénario précédent, le

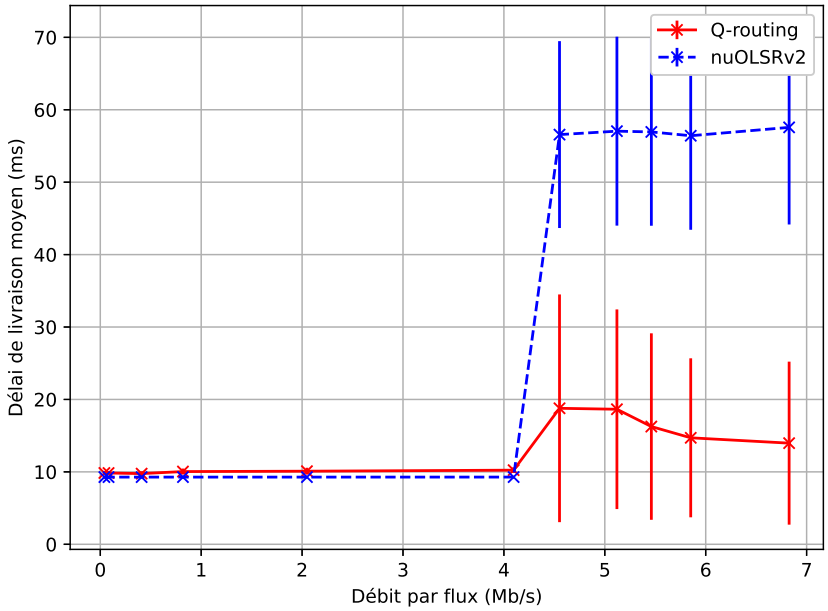


FIG. 5.10 : Délai de livraison moyen pour l'ensemble des sources en fonction du débit sur la grille irrégulière.

profil des courbes est différent.

À faible charge, Q-routing offre des performances inférieures mais comparables à celles de nuOLSRv2. À forte charge et en présence de chemins alternatifs, la situation devient plus favorable à Q-routing. Il offre ainsi en moyenne le meilleur taux de paquets livrés et le délai de livraison le plus court. Ces résultats peuvent s'expliquer notamment par l'équilibrage de la charge réseau entre les chemins entre les nœuds 16 et 22 et entre les nœuds 4 et 34. L'alternance du trafic n'est pas favorable à Q-routing. En effet, Q-routing a une stratégie gloutonne. Il peut abandonner définitivement un chemin si la Q-valeur associée est plus élevée que celles des autres chemins. Lorsque le débit est élevé, ce cas peut se produire plus facilement.

5.4 Conclusions

Dans ce chapitre, nous avons vu en détails notre implémentation de Q-routing. Pour notre implémentation de Q-routing, nous avons réutilisé partiellement une implémentation basique de Bellman-Ford. Nous avons dû faire certains choix concernant l'implémentation. Le premier concerne le type de routage utilisé. Nous avons opté pour un routage proactif. Le deuxième point est la mesure du délai. Nous avons préféré utiliser les paquets de routage pour effectuer les mesures du délai. Cela implique deux hypothèses sur le fonctionnement des nœuds : l'utilisation d'une seule file d'attente et la synchronisation des horloges des nœuds. En revanche, cette méthode permet de limiter

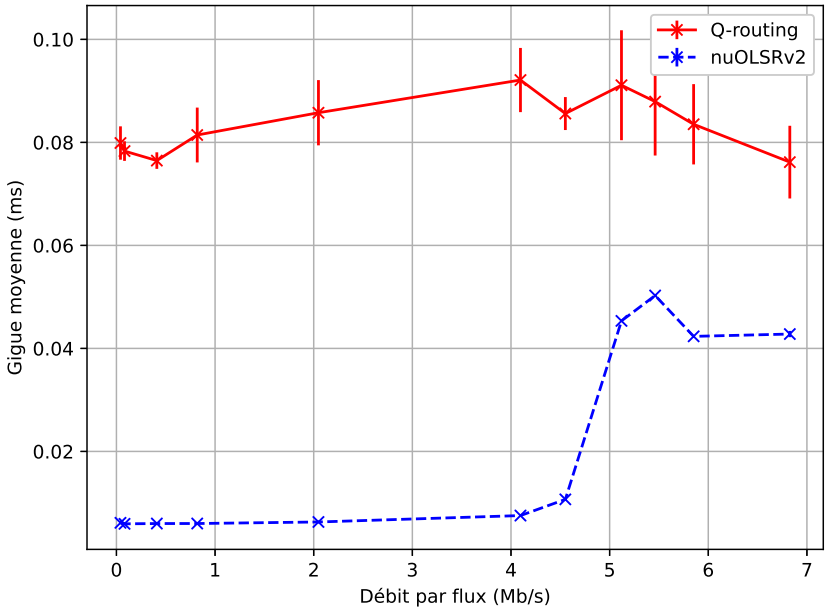


FIG. 5.11 : Gigue moyenne pour l'ensemble des sources en fonction du débit sur la grille irrégulière.

le surcoût lié au routage. Enfin, nous avons préféré actualiser les Q-valeurs à chaque mise à jour des routes plutôt qu'à chaque paquet. Ce choix permet aussi d'économiser de la bande passante.

Nous avons évalué notre implémentation dans deux scénarios différents. OLSRv2 est notre référence car c'est un protocole de routage standardisé et spécialisé dans les réseaux ad-hoc. De plus, OLSRv2 utilise des fonctions auxiliaires, contrairement à notre implémentation. Le premier scénario montre que Q-routing est capable d'utiliser un chemin alternatif même s'il est plus long en cas de congestion. Q-routing obtient donc des meilleurs résultats qu'OLSRv2. Sur la grille irrégulière, les résultats sont plus contrastés. À bas et moyen débit, les performances de Q-routing sont certes similaires mais restent inférieures à celles d'OLSRv2. À débit élevé (à partir de 4,5 Mb/s par flux), Q-routing profite des chemins alternatifs pour mieux répartir la charge. Cela permet à Q-routing de livrer plus de paquets et plus rapidement qu'OLSRv2. Dans les deux scénarios, la gigue moyenne est plus élevée avec Q-routing. À l'exception de la gigue, les résultats de Q-routing sont bons, surtout à charge élevée. Nous verrons ensuite les améliorations que nous avons apporté à Q-routing afin d'atténuer les problèmes apparus en cas de congestion brève.

Chapitre 6

Atténuation des effets de la stratégie gloutonne de Q-routing

Sommaire

7.1 Introduction **105**
7.2 Mise en évidence des problèmes d'estimation **106**
7.2.1 Présentation du problème 106
7.2.2 Scénario 106
7.2.3 Résultats 107
7.3 Solutions proposées **109**
7.3.1 Filtrage des mesures de délais 109
7.3.2 Route à trois statuts 109
7.4 Évaluation des solutions proposées **109**
7.4.1 Scénario avec mobilité 110
7.4.2 Grille sans-fil inspirée de Boyan et Littman 113
7.5 Conclusions **115**

6.1 Introduction

Nous avons vu dans le chapitre précédent l'implémentation originale de Q-routing. Nous l'avons testé sur deux scénarios. Le premier scénario montrait la capacité de Q-routing à utiliser un chemin alternatif bien que plus long. Ce scénario est par nature avantageux pour Q-routing. Les résultats montrent que Q-routing offre de meilleures performances que nuOLSRv2 en termes de taux de paquets livrés et en délai de livraison. Le deuxième scénario utilise la grille irrégulière de Boyan et Littman. 4 flux « verticaux » et 4 flux « horizontaux » apparaissent alternativement sur la grille. Dans

ces conditions, Q-routing n'offre ses meilleures performances que lorsque l'utilisation des chemins alternatifs devient avantageuse. À faible débit, nuOLSRv2 offre un meilleur taux et un meilleur délai de livraison. Nous avons eu plusieurs interrogations sur le comportement de Q-routing. Comme le précise [BL94], Q-routing utilise une stratégie gloutonne, c'est-à-dire qu'il va toujours choisir la route ayant la Q-valeur la plus faible et seules les routes actives sont mises à jour. D'après cette stratégie, Q-routing n'utilisera plus une route si une congestion apparaît momentanément sur celui-ci et si sa Q-valeur n'est plus la plus faible. Dans ce chapitre, nous commencerons par mettre en évidence cet effet de la stratégie gloutonne (Section 6.2). Nous proposerons ensuite deux solutions pour atténuer l'effet de la stratégie gloutonne de Q-routing (Section 6.3). Enfin, nous évaluerons nos solutions avec quatre scénarios sur deux topologies sans-fil (Section 6.4).

6.2 Mise en évidence des effets de la stratégie gloutonne

Q-routing utilise une stratégie gloutonne, c'est-à-dire qu'il va toujours choisir la route ayant la Q-valeur la plus faible et seules les routes actives sont mises à jour. Cette stratégie pose problème surtout en cas de forte variation du délai. En effet, une forte mais brève hausse du délai pourrait augmenter la Q-valeur de la route concernée. Q-routing n'utilisera plus une route si une congestion apparaît momentanément sur celui-ci et si sa Q-valeur n'est plus la plus faible. Cette propriété peut priver Q-routing presque définitivement d'une route si la congestion est intense.

Dans cette section, nous mettons en évidence le principal inconvénient de la stratégie gloutonne de Q-routing. Pour cela, nous allons utiliser une topologie proche de notre précédent exemple jouet, illustré par la Figure 6.1. Il a la particularité d'utiliser des liens sans-fil. Nous utilisons IEEE 802.11a car son fonctionnement est proche des amendements plus récents et qu'il est possible de définir une capacité du lien. Les liens ont une capacité de 9 Mb/s afin qu'il soit saturé plus facilement. La couche liaison IEEE 802.11e est la plus récente disponible et adaptée à cette utilisation. Dans ce scénario, le flux principal de données est situé entre le nœud source 1 et le nœud destination 4. La source de trafic principale commence à émettre de la première minute à l'avant-dernière minute à un rythme d'un paquet toutes les 10 ms soit 410 kb/s. Le trafic d'arrière-plan apparaît entre les nœuds 2 et 3 au bout de 8 minutes de temps simulé et s'arrête au bout d'une seconde. Le débit du trafic d'arrière-plan est de 5,1 Mb/s. Nous relevons le nombre de sauts moyen toutes les minutes au cours de la simulation. Cette métrique varie lentement. La durée simulée est donc de 30 minutes afin d'obtenir un résultat visible. Le Tableau 6.1 récapitule les paramètres de simulation. Les simulations sont réalisées avec le simulateur réseau Qualnet 8.2. L'objectif est d'utiliser le nombre de sauts moyen pour déterminer la route utilisée.

La Figure 6.2 illustre le nombre de sauts moyen en fonction du temps simulé. Entre le début de l'émission de données et 8 minutes, le flux de données effectue trois sauts pour atteindre sa destination. L'écart-type se réduit. Le flux de données est acheminé par le chemin le plus court. La présence de quelques valeurs éloignées proche de 4 indique que dans de rare cas, Q-routing commence par utiliser le chemin alternatif plus long. Puis, il finit par utiliser le chemin le plus court. À partir de 8 minutes, le nombre de sauts moyen augmente progressivement jusqu'à la fin de la

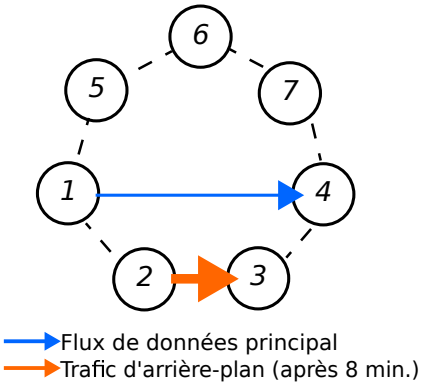


FIG. 6.1 : Topologie de mise en évidence. Les nombres correspondent aux identifiants des nœuds.

Élément	Paramètre	Valeur
Simulation	Graines	30 graines différentes
Réseau	Lien	IEEE 802.11a 9 Mb/s couche liaison IEEE 802.11e
	Topologie	Anneau
Nœud	File d'attente	1 file FIFO
Données applicatives	Taille par envoi	512 octets
	Début	1 min
	Fin	1 min avant la fin de simulation
Flux d'arrière-plan	Taille par envoi	512 octets
	Début	8 min.
Q-routing	η	0,9
	Phase de découverte	15 s

TAB. 6.1 : Paramètres de simulation.

simulation. L'écart-type est très faible, ce qui indique que Q-routing utilise le chemin alternatif dans la majorité des cas. Quelques valeurs éloignées montrent que Q-routing peut rester sur le chemin le plus court dans de rares cas.

Le nœud 1 utilisait le chemin le plus court (via les nœuds 2 et 3) jusqu'à la huitième minute. Lorsque notre trafic d'arrière-plan apparaît pendant juste une seconde, le nœud 1 change de route. Il utilise le chemin alternatif plus long (via les nœuds 5, 6 et 7). Après cette congestion momentanée, Q-routing continue d'utiliser le chemin alternatif même s'il est plus lent. La route la plus courte n'a plus la plus petite Q-valeur, car sa Q-valeur a fortement augmenté durant la congestion. Elle n'est plus sélectionnée et ne peut plus être sélectionnée tant que les autres ont une Q-valeur plus petite. Avec ce premier scénario, nous venons de confirmer expérimentalement que Q-routing est sensible aux optimums locaux. Cette caractéristique peut se révéler gênante notamment lorsque le délai devient difficile à estimer correctement ou que les conditions de trafic et de topologie changent fré-

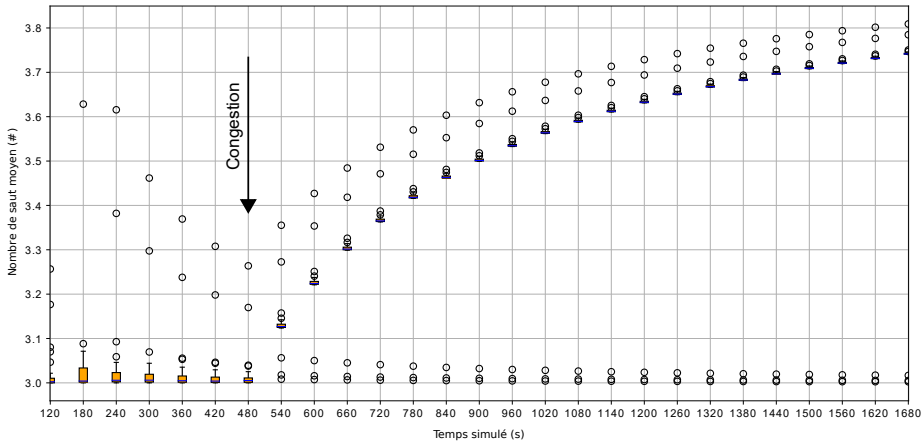


FIG. 6.2 : Nombre de sauts moyen en fonction du temps simulé dans notre scénario de mise en évidence du caractère glouton de Q-routing.

quemment. Par exemple en mobilité, les changements de routes peuvent être fréquents et toutes les routes doivent être mises à jour fréquemment.

6.3 Solutions proposées

Q-routing est donc un algorithme glouton. Nous avons vu qu'une seconde de congestion peut priver Q-routing du chemin le plus court et le plus rapide. Dans cette section, nous présentons trois solutions pour atténuer le comportement glouton de Q-routing : la stratégie ε-gloutonne, les mises à jour ε-gloutonnes et le système à réinitialisation périodique des Q-valeurs.

6.3.1 Approche probabiliste inspirée de l'apprentissage par renforcement

Stratégie ε-gloutonne

Nous avons vu que pour qu'une route soit mise à jour, il faut que la route soit sélectionnée. Dans la stratégie gloutonne, il faut que la route ait la plus petite Q-valeur parmi les autres routes candidates. Dans le scénario précédent, après une seconde de congestion, la route la plus courte n'a plus la plus petite Q-valeur. Elle n'est plus sélectionnée et ne peut plus être sélectionnée tant que les autres ont une Q-valeur plus petite. En apprentissage par renforcement, c'est un problème connu d'exploration-exploitation [SB18]. Une des solutions est l'utilisation d'une stratégie ε-gloutonne. Avec cette stratégie, une route peut être sélectionnée si elle a la plus petite Q-valeur. Mais elle a aussi une probabilité ε d'être choisie indépendamment de sa Q-valeur.

Cette solution est simple à mettre en œuvre néanmoins elle a quelques inconvénients. La qualité de la route n'est pas garantie dans le cas où elle n'a pas la Q-valeur minimale. C'est-à-dire que dans le pire cas, la pire route disponible est choisie. Dans notre cas précédent, cela pourrait être aussi bien

le chemin le court que le plus long. Un deuxième inconvénient est que cette stratégie favorise les boucles de routage. Par exemple, deux nœuds peuvent se choisir mutuellement comme prochain saut. Avec une stratégie gloutonne, ce cas est moins probable à cause de la minimisation de la Q-valeur et donc de la minimisation implicite du délai.

Mise à jour ε-gloutonne

Nous avons vu que la stratégie ε est simple à appliquer mais elle n'est pas parfaite. En effet, il n'y a pas de garantie concernant la qualité de la route lorsqu'elle choisit aléatoirement. Cependant, il est possible de l'adapter pour mettre à jour les Q-valeurs sans pour autant choisir une route aléatoirement. En effet, au lieu de choisir une route avec une probabilité ε, nous proposons de mettre à jour la Q-valeur d'une route avec la probabilité ε. La Q-valeur est déjà mise à jour si la route est sélectionnée. Ainsi, les Q-valeurs des routes non-utilisées sont mises à jour suivant une probabilité ε.

6.3.2 Système d'époque

La dernière solution proposée est inspirée des cycles d'apprentissage en apprentissage automatique. Il s'agit d'un système à réinitialisation périodique. Un cycle est composé de deux phases : une phase de découverte et une phase d'exploitation. Au début de la phase de découverte, une nouvelle Q-table est créée. La Q-table désigne l'ensemble des paramètres et valeurs prises par la Q-fonction. Il peut s'agir soit d'une copie de l'ancienne table dont les Q-valeurs ont réinitialisées, soit d'une table vide. Dans le cas d'une table vide, les Q-valeurs des routes ajoutées par la suite sont à 0 (à cause de l'initialisation optimiste). Pendant la phase de découverte du nouveau cycle, l'ancienne Q-table est toujours utilisée. Cela permet une transition plus douce entre chaque cycle. Les routes de la deuxième table sont mises à jour suivant les mêmes règles. Comme aucune route n'est choisie dans la deuxième table, leur Q-valeur n'est donc pas mise à jour. Pour autant, les données (comme le délai) de la route sont bien mises à jour. A la fin de la phase de découverte, les Q-valeurs de toutes les routes de la deuxième table sont mises à jour. La première table est détruite et la seconde table remplace la première. Pour chaque destination. Il y a un nouveau choix de route.

En dehors de l'avantage d'atténuer le comportement glouton, ce système permet de purger régulièrement les routes périmées. Cependant, il y a aussi quelques inconvénients. Le premier inconvénient est le doublement temporaire du besoin en mémoire. En effet, durant la phase de découverte, deux Q-tables sont nécessaires en même temps. La difficulté de paramétrage de la durée des phases de découverte et d'exploitation est le second inconvénient. En effet, si la durée de découverte est trop courte, les Q-valeurs n'auront pas été suffisamment mises à jour et ne refléteront pas la réalité. Si la phase d'exploitation est trop courte, Q-routing sera trop fréquemment en phase de découverte. Enfin, la durée d'un cycle ne doit pas être trop longue, sinon le système de réinitialisation perd de son intérêt. En soit, cette approche est proche conceptuellement de PQ-routing de Choi et Yeung [CY96]. En effet, PQ-routing utilise des tables supplémentaires permettant de res-

taurer la Q-table à un état antérieur. Contrairement à PQ-routing, notre Q-routing avec époques réinitialise la Q-table périodiquement et n'utilise pas le plus court chemin à faible charge.

6.4 Évaluation des solutions proposées

Dans cette section, nous proposons d'évaluer les propositions précédentes dans divers scénarios. L'ensemble des résultats a été obtenus avec Qualnet 8.2.

6.4.1 Exemple jouet sans-fil

Pour évaluer nos deux solutions, nous allons commencer avec deux scénarios simples. Le premier scénario est le scénario précédent décrit dans la Section 6.2. L'objectif est d'évaluer si nos solutions permettent de retourner sur le chemin le plus court. Le second scénario reprend notre exemple jouet du chapitre précédent (voir la sous-section 5.3.1) mais en utilisant une topologie sans-fil. Nous allons utiliser une topologie sans-fil illustrée par la Figure 6.1. Le Tableau 6.1 récapitule les paramètres utilisés pour les deux scénarios.

Une perturbation d'une seconde

Nous reprenons le scénario de mise en évidence du comportement glouton de la Section 6.2. Cette fois, nous allons évaluer séparément Q-routing avec époques (sous-section 6.3.2) et Q-routing avec mises à jour ε-gloutonnes (sous-section 6.3.1). La phase d'exploitation d'une époque est de 300 secondes. Nous choisissons une valeur ε de 10 %. Il s'agit d'un bon compromis entre la probabilité de mise à jour et les résultats obtenus (non présentés) sur le scénario de la Section 6.4.1. L'objectif est de caractériser le comportement de chacune de nos solutions proposées.

La Figure 6.3 illustre le nombre de sauts moyen en fonction du temps simulé. Le nombre de sauts moyen augmente à partir de 8 minutes comme dans les cas précédents. Contrairement à Q-routing avec époques, la médiane du nombre de sauts moyen continue d'augmenter. Q-routing avec mises à jour ε-gloutonnes agit comme Q-routing selon la valeur médiane du nombre de sauts. À partir de l'écart-type, nous déduisons que Q-routing avec époques peut réutiliser le chemin le plus court après la congestion. Mais ce cas dépend finalement de l'aléatoire. Car dans notre cas, Q-routing avec mises à jour ε-gloutonnes n'a que 10 % de chance toutes les 10 s de mettre à jour la route passant par le chemin le plus court. La mise à jour des Q-valeurs suit donc en partie une approche probabiliste. Q-routing avec mises à jour ε-gloutonnes permet d'atténuer le comportement mais il nécessite du temps ou de la chance pour être efficace.

La Figure 6.4 montre le nombre de sauts moyen en fonction du temps simulé. Le nombre de sauts moyen semble plus élevé entre 120 s et 480 s qu'avec la version originale de Q-routing. Cela peut être en partie expliqué par le changement d'époque autour de 300 s. Comme pour la version originale de Q-routing, le nombre de sauts moyen augmente à partir de 8 minutes (480 s). Cela est dû à la congestion. À partir 660 s, le nombre de sauts moyen commence à diminuer. Ce moment

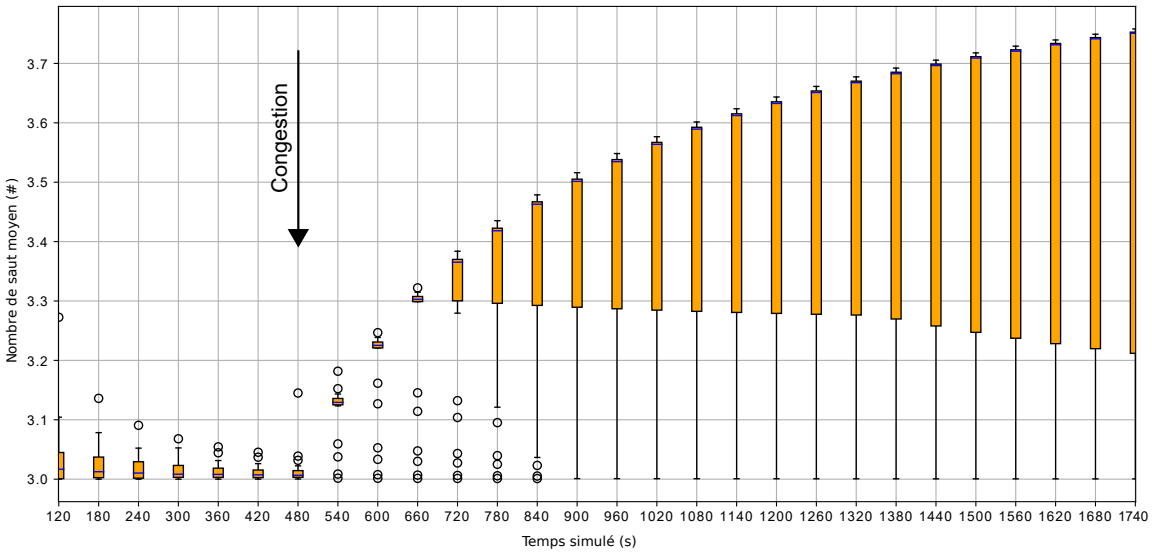


FIG. 6.3 : Nombre de sauts moyen en fonction du temps simulé pour Q-routing avec mises à jour ε-gloutonne.

équivaut au début de la troisième époque. En effet, chaque époque est composée de 15 s de découverte et 300 s d'exploitation. Q-routing avec époques est efficace. Il permet d'actualiser rapidement toutes les Q-valeurs. Cependant, il est plus difficile à paramétrer finement.

Conclusion du premier scénario

Dans ce premier scénario, nous évaluions la capacité de nos solutions à retourner sur le chemin le plus court. Les résultats montrent une différence entre nos propositions. En effet, Q-routing avec époques permet de revenir sur le chemin le plus court en 200 s (à t=600 s). Cette durée dépend néanmoins des paramètres de Q-routing avec époques. Q-routing avec mises à jour ε-gloutonnes nécessite plus de temps pour revenir sur le chemin le plus court (via les nœuds 2 et 3). Q-routing avec époques est donc plus efficace que Q-routing avec mises à jour ε-gloutonnes. La mise à jour des Q-valeurs suit en partie une approche probabiliste pour Q-routing avec mises à jour ε-gloutonnes. Son efficacité est donc conditionnée par l'aléatoire du simulateur. L'efficacité de Q-routing avec époques dépend seulement des paramètres de celui-ci. Pour les prochains scénarios, nous évaluerons donc uniquement Q-routing, Q-routing avec époques et nuOLSRv2.

Exemple jouet sans-fil

Pour ce second scénario, nous reprenons en partie l'exemple jouet du chapitre précédent (Section 5.3.1). Le trafic d'arrière-plan s'arrête à l'avant-dernière minute de la simulation pour éviter d'éventuels biais liés à la simulation. La simulation dure 15 minutes. Nous évaluons Q-routing, Q-

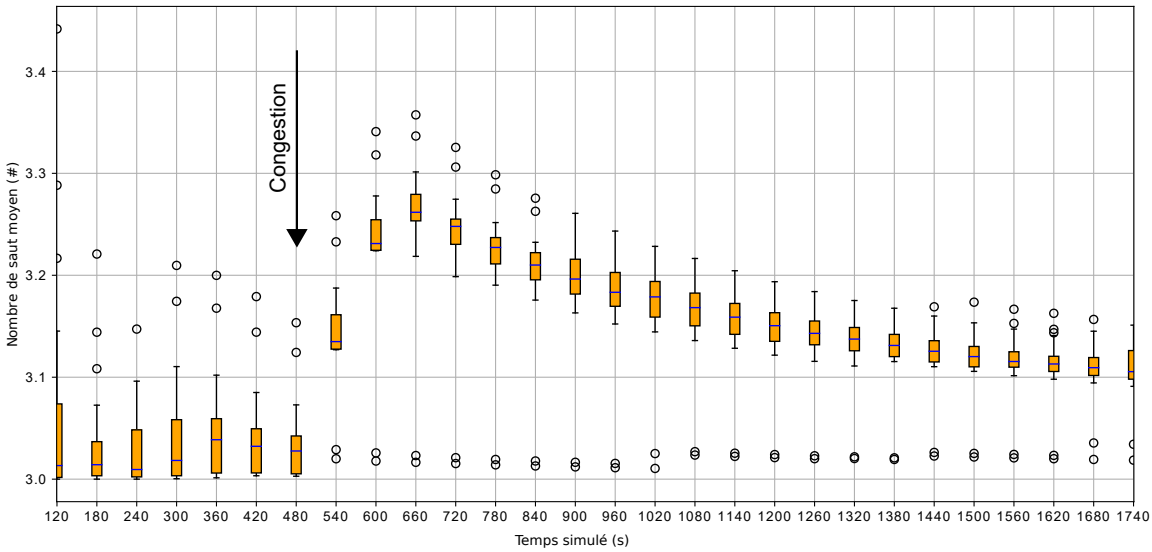


FIG. 6.4 : Nombre de sauts moyen en fonction du temps simulé pour Q-routing avec époques. Représentation boîte à moustache.

routing avec époques et nuOLSRv2. La phase d'exploitation de Q-routing avec époques dure 300 s comme pour le scénario précédent. L'objectif est d'évaluer la capacité de Q-routing à utiliser un chemin alternatif sur une topologie sans-fil simple. Le second objectif est de fournir une première évaluation des performances de Q-routing avec époques.

La Figure 6.5 représente le taux de paquets livrés en fonction du débit d'arrière-plan. Nous rappelons que les paquets désordonnés sont détruits par la destination. Jusqu'à 3,9 Mb/s, Q-routing et nuOLSRv2 délivrent la même quantité de paquets. Q-routing avec époques perd en moyenne un peu plus de paquets par rapport à Q-routing original. Entre 3,8 et 4,5 Mb/s, le taux de paquets reçus diminue de 10 % pour les trois protocoles. En moyenne, nuOLSRv2 offre le meilleur taux de paquets livrés sur ce segment. À partir 4,5 Mb/s, le taux de paquets livrés de nuOLSRv2 plonge. Étonnamment, Q-routing avec époques a le même profil que nuOLSRv2. À partir de 4,8 Mb/s, Q-routing avec époques livre plus de paquets que nuOLSRv2. En moyenne, son taux de paquets livrés est jusqu'à 10 % supérieur à celui de Q-routing avec époques et jusqu'à 12 % supérieur par rapport à celui de nuOLSRv2. Néanmoins, l'écart-type de la version originale de Q-routing est très important.

La Figure 6.6 illustre le délai de bout-en-bout. Jusqu'à 3,5 Mb/s, le délai de livraison est similaire pour les trois protocoles. Entre 3,4 et 3,56 Mb/s, le délai de livraison augmente surtout avec nuOLSRv2 et Q-routing avec époques. Le délai avec Q-routing avec époques reste 5 fois moins élevé que celui de nuOLSRv2. Pour Q-routing, le délai ne change peu sur l'intervalle. Entre 3,56 Mb/s et 3,8 Mb/s, le délai de livraison diminue fortement avec nuOLSRv2 et Q-routing avec époques. Entre 3,8 et 3,9 Mb/s, le délai de livraison augmente pour les trois protocoles. Q-routing atteint son pic à 60 ms et Q-routing avec époques à 80 ms. Avec nuOLSRv2, les données transitent sur le réseau pendant 120 ms. Il n'y a pas de changement notable sur l'intervalle 3,9 - 4,5 Mb/s. En moyenne,

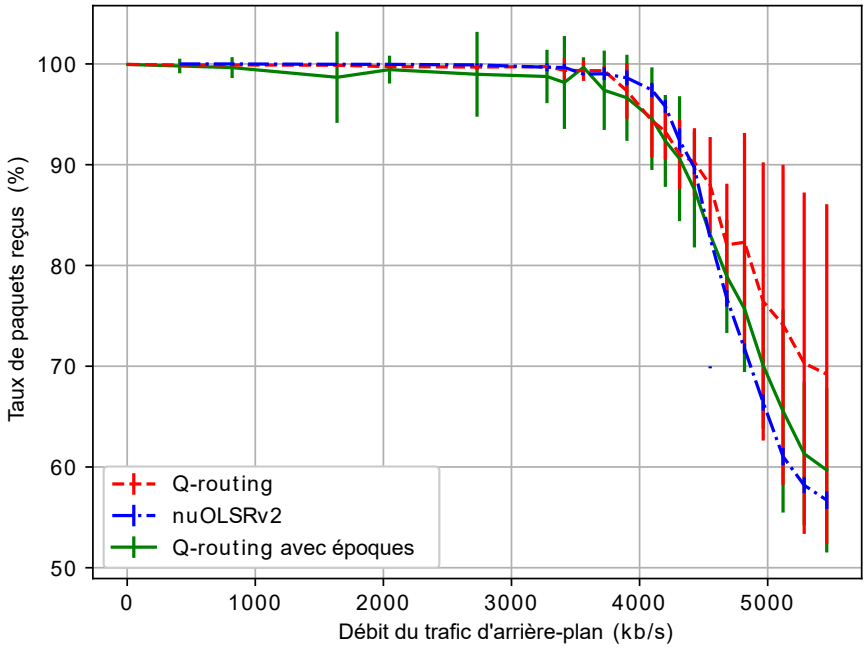


FIG. 6.5 : Taux de paquets livrés en fonction du débit d'arrière-plan avec notre scénario jouet sans-fil.

Q-routing offre le meilleur délai de livraison sur l'ensemble des débits testés, bien que l'écart-type soit important.

La Figure 6.7 montre la gigue en fonction du débit d'arrière-plan. Jusqu'à 3,5 Mb/s, les trois protocoles offrent une gigue comparable. Entre 3,4 et 3,56 Mb/s, la gigue augmente fortement pour nuOLSRv2 et Q-routing avec époques. Cela correspond à un pic de délai de livraison pour ces deux protocoles. L'augmentation est plus contenue avec Q-routing. Entre 3,56 et 3,8 Mb/s, la gigue diminue pour nuOLSRv2 et Q-routing avec époques. Entre 3,8 et 4,2 Mb/s, la gigue augmente pour les trois protocoles. Q-routing atteint un pic de gigue à 1,2 ms, soit le double de nuOLSRv2 à ce débit. Sur l'ensemble des débits, nuOLSRv2 offre en moyenne la meilleure gigue. Q-routing avec époques a étonnamment le même profil de courbe que nuOLSRv2.

Sur l'ensemble des trois métriques, nuOLSRv2 et Q-routing avec époques ont globalement le même profil de courbe. En termes de taux de paquets livrés et de délai de livraison, Q-routing offre en moyenne les meilleures performances. Ces résultats restent à prendre avec précaution car l'écart-type est important. Les performances de Q-routing avec époques sur ce scénario sont moins bonnes que celles de la version originale. Cette contre-performance peut être liée à notre modification. Lorsque Q-routing avec époques renouvelle sa Q-table à la fin de la phase de découverte, les Q-valeurs n'ont pas été assez actualisées pour refléter l'état du réseau et notamment sa congestion. En effet, la phase de découverte ne dure que 15 s, ce qui correspond à une à deux mises à jour.

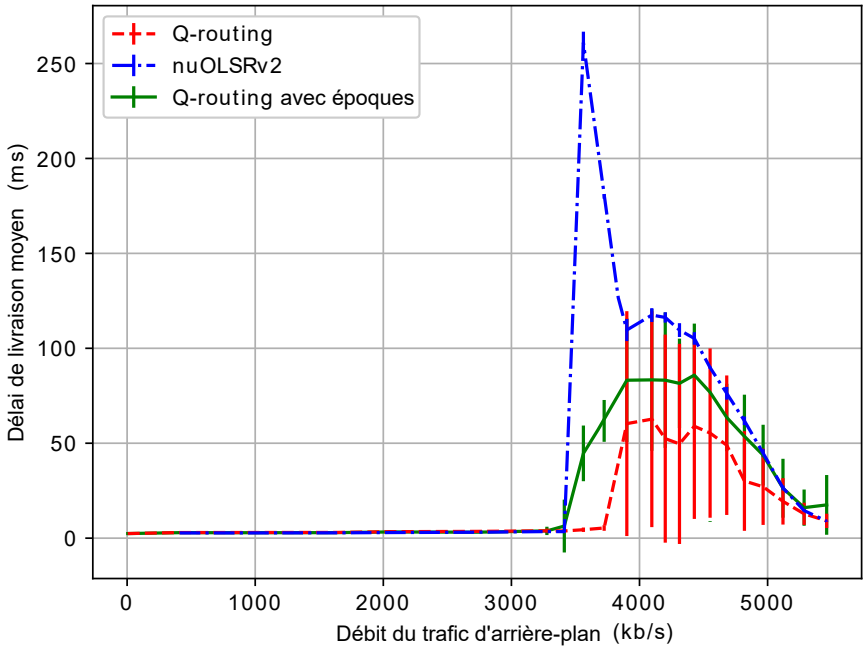


FIG. 6.6 : Délai de livraison moyen en fonction du débit d'arrière-plan avec notre scénario jouet sans-fil.

6.4.2 Grille sans-fil inspirée de Boyan et Littman

Nous allons maintenant évaluer Q-routing avec époques sur une topologie plus grande : la version sans-fil et adaptée de la grille de Boyan et Littman. La grille irrégulière de Boyan et Littman ne peut pas être transformée en topologie sans modification. Nous avons pour cela enlever les nœuds 11, 16, 23 et 29. Nous utilisons des liens sans-fil IEEE 802.11a car son fonctionnement est proche des amendements plus récents et qu'il est possible de définir une capacité du lien. Les liens ont une capacité de 9 Mb/s afin qu'il soit saturé plus facilement. La couche liaison IEEE 802.11e est la plus récente disponible et est adaptée pour cette utilisation. La Figure 6.8 illustre cette topologie. Dans le premier scénario, le trafic sera localisé sur une partie de la grille. L'objectif est de voir si Q-routing et Q-routing avec époques sont capables d'utiliser un chemin alternatif sur une topologie plus grande. Dans ce second scénario, les couples sources - destinations seront répartis aléatoirement sur la grille et dans le temps. Le but est de vérifier que Q-routing et Q-routing avec époques fonctionnent sur une grande topologie sans-fil et avec un trafic diffus. Le Tableau 6.2 récapitule les paramètres communs aux deux scénarios. Pour les deux scénarios, nous évaluons Q-routing, Q-routing avec époques et nuOLSRv2.

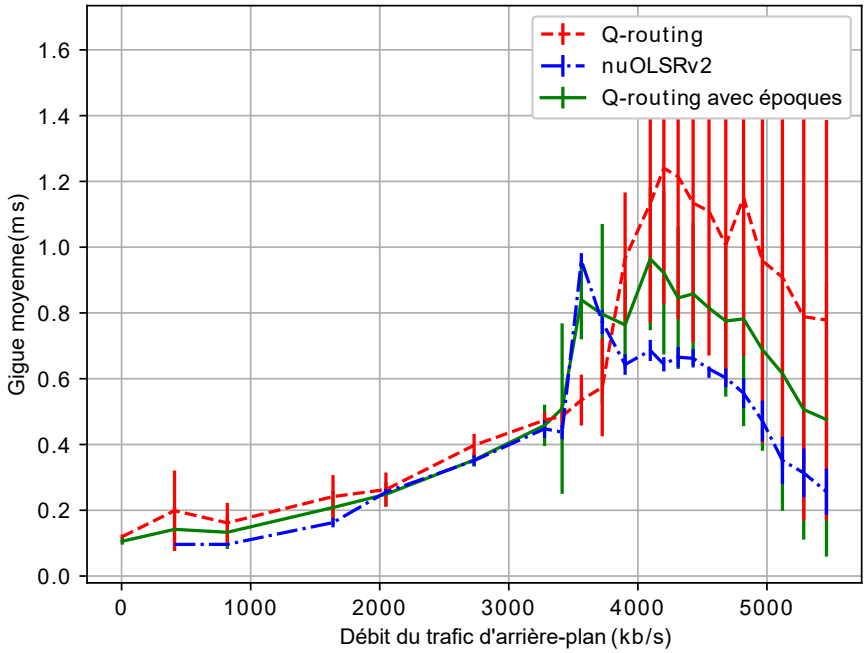


FIG. 6.7 : Gigue moyenne en fonction du débit d'arrière-plan avec notre scénario jouet sans-fil.

Trafic localisé

Dans ce premier scénario, le trafic est localisé surtout sur la moitié supérieure de la grille. Il y a 4 flux de données. Le Tableau 6.3 indique l'emplacement des couples sources - destinations. Les sources émettent entre la première et l'avant-dernière minute de simulation pour éviter d'éventuels biais de simulation. Les sources sont synchronisées et ont le même débit. Elles émettent leurs paquets au même instant. L'objectif de ce scénario est de vérifier que Q-routing est capable répartir la charge sur un chemin alternatif sur une topologie plus grande que notre exemple jouet. Pour réussir cet objectif, il faut remplir deux conditions. Premièrement, le flux entre les nœuds 35 - 5 doit utiliser la route passant par 36 et 6. Pourtant, il existe deux routes de même distance entre 35 et 5 donc un protocole de routage utilisant le plus court chemin n'est pas sûr de remplir cette condition. La deuxième condition est que le flux entre 4 et 34 emprunte aussi la route passant par 36 et 6. L'utilisation de cette route ne correspond pas au plus court chemin. La simulation dure 15 minutes.

La Figure 6.9 illustre le taux de paquets livrés en fonction du débit par source de trafic. Le taux de paquets reçus chute rapidement avec nuOLSRv2. À 400 kb/s par source, seulement la moitié de paquets arrive à destinations. À 700 kb/s par source, il ne reste plus que 20 % de paquets. À partir de 700 kb/s par source, le taux de paquets reçus diminue certes, mais plus lentement. Q-routing et Q-routing avec époques ont le même profil de courbe. Le taux de paquets livrés diminue lentement jusqu'à 400 kb/s par source. Les deux versions de Q-routing livrent alors 95 % des paquets. Le taux de paquets livrés chute fortement d'environ 50 % entre 400 kb/s et 800 kb/s par source. À partir de

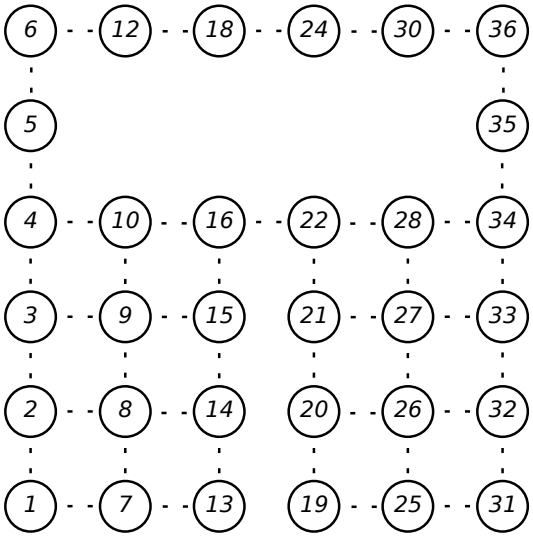


FIG. 6.8 : Grille sans-fil adaptée de la grille de Boyan et Littman

Élément	Paramètre	Valeur
Simulation	Graines	30 graines différentes
Réseau	Lien	IEEE 802.11a 9 Mb/s couche liaison IEEE 802.11e
	Topologie	grilles
Nœud	File d'attente	1 file FIFO
Données applicatives	Taille par envoi	512 octets
	Début	1 min
	Fin	1 min avant la fin de simulation
Flux d'arrière-plan	Taille par envoi	512 octets
	Début	8 min.
Q-routing	η	0,9
	Phase de découverte	45 s
	Phase d'exploitation	300 s

TAB. 6.2 : Paramètres de simulation pour la grille sans-fil.

1250 kb/s par source, le taux de paquets reçus diminue plus lentement, mais il ne reste que 20 % de paquets livrés. Les deux versions de Q-routing offrent de meilleures performances que nuOLSRv2 en termes de paquets livrés indépendamment du débit. Q-routing avec époques offre même une légère amélioration (autour de 5 % à 600 kb/s).

La Figure 6.10 représente le délai de livraison moyen en fonction du débit par source de trafic. Comme pour le taux de paquets livrés, Q-routing et Q-routing avec époques ont le même profil de

Source	Destination
35	5
4	34
9	27
26	8

TAB. 6.3 : Emplacement des flux de données sur la grille (trafic localisé).

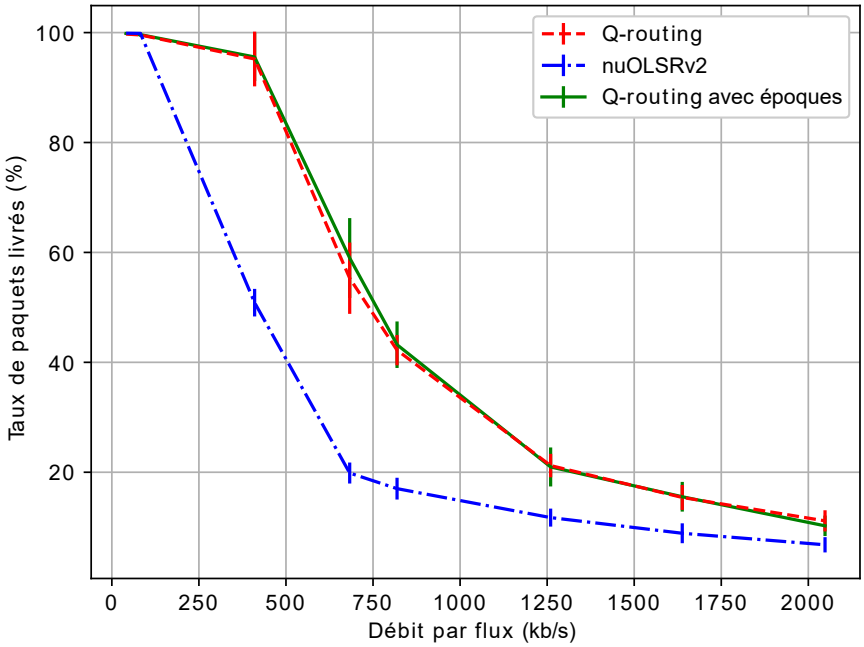


FIG. 6.9 : Taux de paquets reçus pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic localisé).

courbe. Le délai de livraison augmente rapidement avec nuOLSRv2. À 400 kb/s, le délai de livraison est 7 s. Au même débit, les deux versions de Q-routing livrent les paquets en moins d'une seconde. Néanmoins, à partir de 400 kb/s, le délai se dégrade rapidement aussi pour les Q-routing. Entre 800 et 1250 kb/s par source, les Q-routing deviennent plus lents que nuOLSRv2. Q-routing est plus lent mais livre plus de paquets à débit élevé. Q-routing avec époques présente une très légère amélioration autour de 1250 kb/s par source.

Enfin, la Figure 6.11 montre la gigue moyenne en fonction du débit par source de trafic. La gigue se dégrade plus rapidement avec nuOLSRv2 qu'avec les deux versions de Q-routing. À débit élevé, la gigue ne dépasse pas 40 ms avec Q-routing alors qu'elle tend vers 100 ms pour nuOLSRv2. Q-routing offre donc la meilleure gigue sur l'ensemble des débits. La gigue est similaire pour les deux versions de Q-routing.

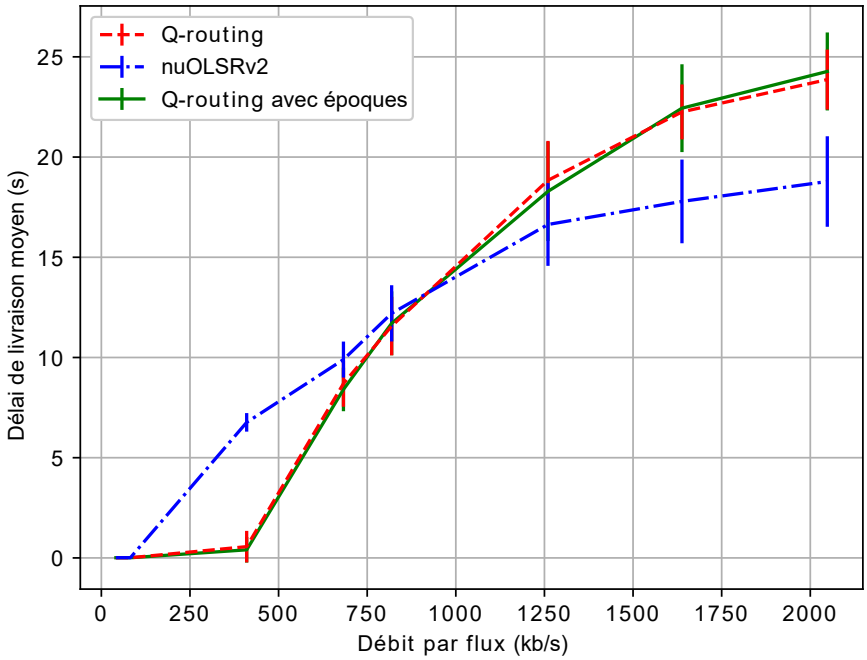


FIG. 6.10 : Délai de livraison moyen pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic localisé).

Trafic diffus

Dans le deuxième scénario, le trafic est réparti sur toute la grille. Il y a 15 sources de trafic réparties aléatoirement sur la grille. Elles émettent de façon synchronisée et ont le même débit. La Figure 6.12 décrit les couples sources - destinations et leur période d'activation. Jusqu'à 5 sources de données peuvent être actives en même temps sur trois intervalles. La simulation dure 30 minutes. L'objectif est de vérifier que Q-routing peut fonctionner sur une grande topologie avec un trafic aléatoire.

La Figure 6.13 représente le taux de paquets livrés en fonction du débit par source de trafic. Les trois protocoles ont le même profil de courbe. Q-routing avec époques offre le meilleur taux de livraison, suit ensuite Q-routing et enfin nuOLSRv2. Le gain apporté par Q-routing avec époques se situe entre 800 kb/s et 2000 kb/s. Au mieux, Q-routing avec époques livre 3 % de paquets de plus que Q-routing à 800 kb/s par source. L'écart avec nuOLSRv2 est de 8 % à 800 kb/s par source. L'écart entre Q-routing et nuOLSRv2 est donc de 5 % à 8 %.

La Figure 6.14 illustre le délai de livraison en fonction du débit par source de trafic. Les trois protocoles ont un profil de courbe similaire. Le délai de livraison se dégrade rapidement pour les trois protocoles entre 400 kb/s et 1250 kb/s par source. À 800 kb/s par source, ils offrent un délai supérieur à 4 s. À 1250 kb/s par source, le délai de livraison de Q-routing avec époques a quasiment doublé. Les paquets ont besoin en moyenne d'au moins 8 s pour atteindre leur destination,

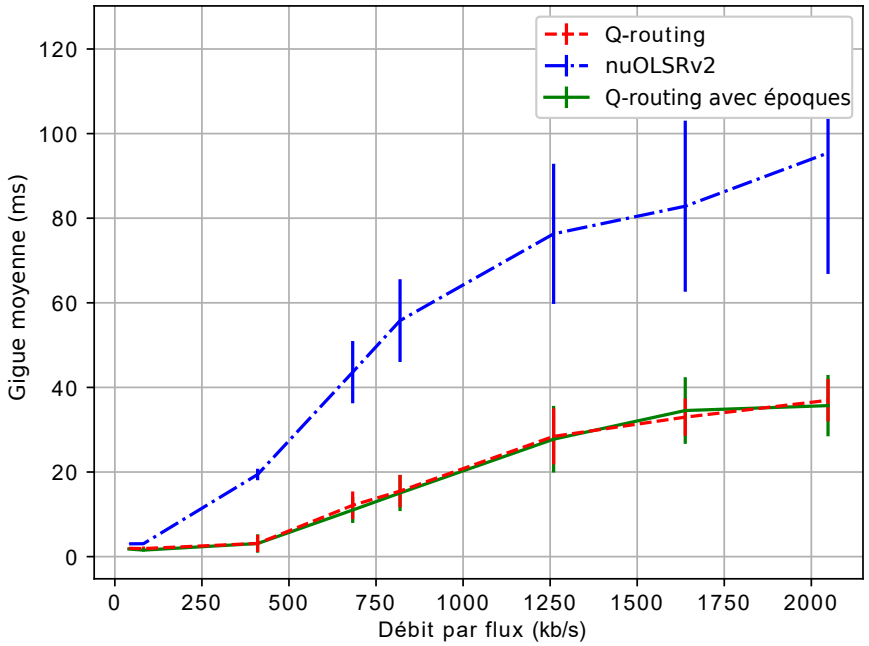


FIG. 6.11 : Gigue moyenne pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic localisé).

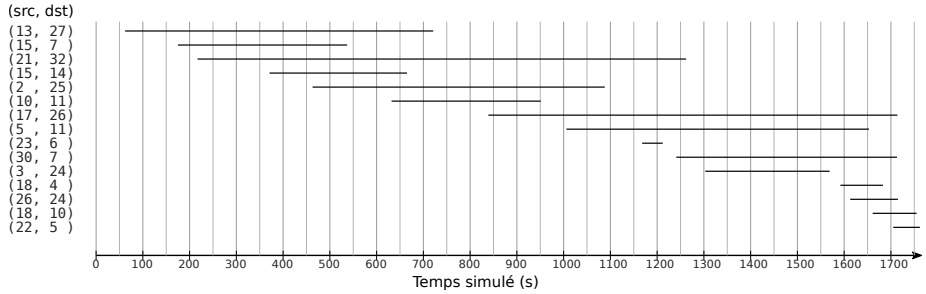


FIG. 6.12 : Chronogramme des flux de données sur la grille irrégulière (trafic diffus).

indépendamment du protocole de routage. L'écart-type est plus important que pour les autres débits pour Q-routing avec époques autour de 2 Mb/s. Q-routing avec époques offre en moyenne le meilleur délai de livraison, suivi de Q-routing et nuOLSRv2. L'écart entre Q-routing avec époques et Q-routing est au mieux d'une demie seconde à 800 kb/s par source.

Enfin, la Figure 6.15 montre la gigue moyenne en fonction du débit par source de trafic. Les deux versions de Q-routing ont le même profil de courbe. La gigue se dégrade plus rapidement avec nuOLSRv2 qu'avec Q-routing. L'écart est de 5 ms à 800 kb/s par source entre Q-routing et nuOLSRv2. Il est de 10 ms à 1,6 Mb/s. Comme pour le délai de livraison, Q-routing avec époques présente une singularité autour de 2 Mb/s. Au point singulier, la gigue est plus élevée pour Q-routing avec époques

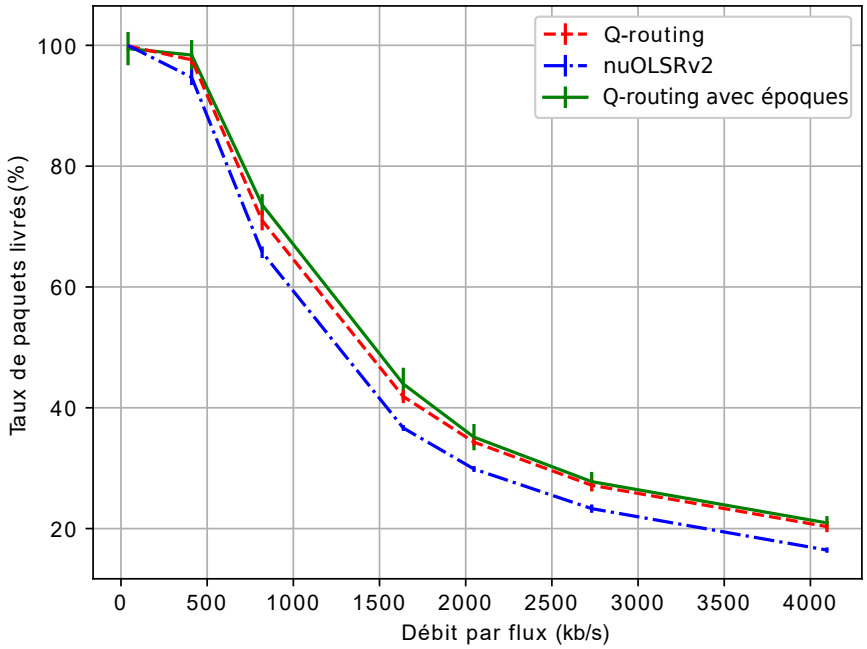


FIG. 6.13 : Taux de paquets reçus pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic diffus).

que pour la version originale. L'écart-type est élevé comparé aux autres points indépendamment du protocole.

Conclusions sur les résultats sur la grille sans-fil

Sur ces deux derniers scénarios, Q-routing avec époques obtient de bons résultats comparés à ceux de nuOLSRv2. En effet, les résultats de Q-routing avec époques sont au moins équivalents voire meilleurs que ceux de la version originale de Q-routing. Ce n'était pas le cas avec la topologie précédente. Q-routing offre de meilleures performances que nuOLSRv2 sur toutes les métriques de qualité de services proposées à l'exception du délai de livraison sur le scénario avec trafic localisé et à forte charge. De plus, la différence de taux de paquets reçus ne permet pas de justifier cette différence de délai. Dans le cas d'un trafic localisé, les deux versions de Q-routing sont capables d'équilibrer la charge entre deux chemins sur une topologie complexe qui est la grille irrégulière sans-fil. Dans le cas d'un trafic diffus, les gains apportés par Q-routing montrent qu'il est capable de fonctionner sur une topologie sans-fil complexe indépendamment de sa localisation.

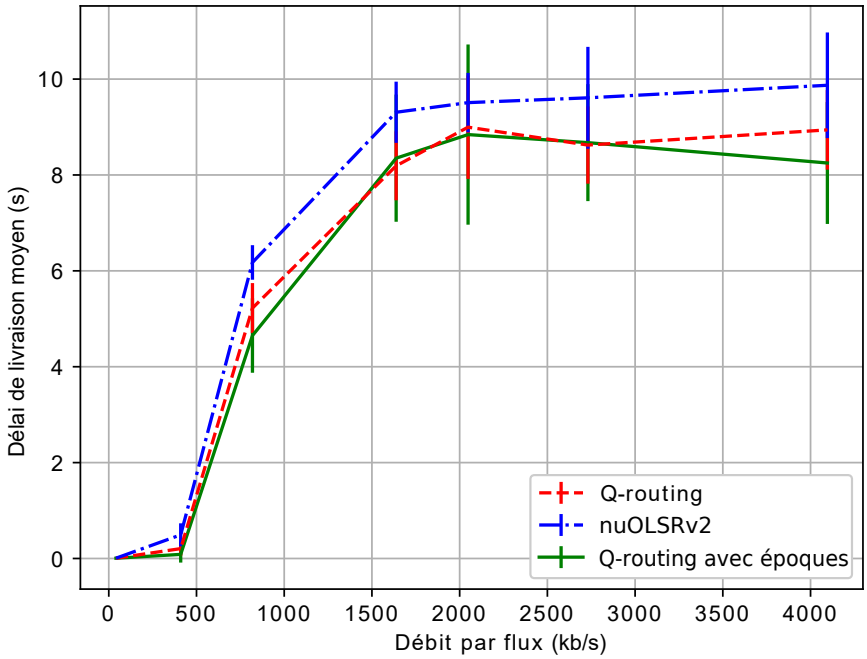


FIG. 6.14 : Délai de livraison moyen pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic diffus).

6.5 Conclusions

Dans ce chapitre, nous avons mis en évidence sur un scénario simple que Q-routing avait un comportement glouton. En effet, en cas de congestion momentanée, Q-routing abandonne le meilleur chemin au profit d'un chemin alternatif. Nous avons proposé trois solutions pour atténuer ce comportement : i) Q-routing avec époques, ii) Q-routing avec stratégie ε-gloutonne et iii) Q-routing avec mises à jour ε-gloutonnes. Nous avons évalué la capacité à réutiliser le chemin le plus court après une congestion pour les deux plus prometteuses : Q-routing avec époques et Q-routing avec mises à jour ε-gloutonnes. Q-routing avec époques est la solution la plus efficace, mais c'est aussi la solution comportant le plus de paramètres. Il est donc plus difficile de trouver un ensemble de paramètres maximisant les performances.

Nous avons évalué Q-routing et Q-routing avec époques sur trois scénarios supplémentaires : un sur notre exemple jouet sans-fil et deux sur une version adaptée de la grille irrégulière de Boyan et Littman. Sur notre exemple jouet sans-fil, les résultats de Q-routing avec époques sont plutôt décevants. Q-routing avec époques livre moins de paquets et plus lentement que la version originale. Les résultats de Q-routing ne sont pas pour autant convaincants. En effet, ils sont proches de ceux de nuOLSRv2 sachant que le scénario est normalement favorable à Q-routing. Sur la topologie sans-fil adaptée de la grille de Boyan et Littman, les résultats de Q-routing sont encourageants, d'autant plus que Q-routing a été créé pour les réseaux filaires. Les performances en termes de taux

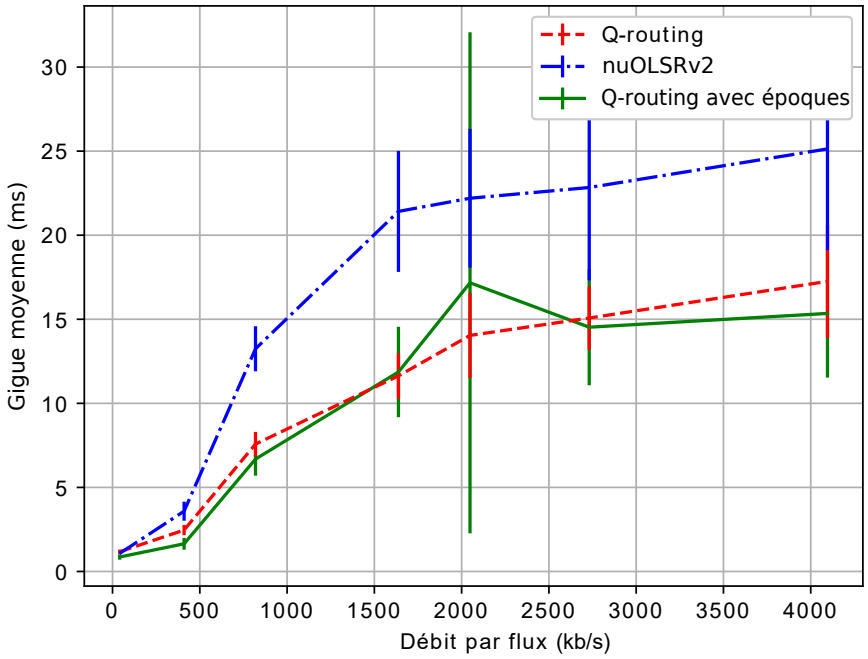


FIG. 6.15 : Gigue moyenne pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic diffus).

de paquets reçus de Q-routing sont plus intéressantes que celles de nuOLSRv2. Pour les deux autres métriques, les résultats dépendent du scénario, voire du débit. Q-routing avec époques obtient des résultats similaires, voire mieux que ceux Q-routing. De plus, Q-routing avec époques atténue le comportement glouton de Q-routing.

Dans le prochain chapitre, nous nous attacherons à l'amélioration de l'estimation du délai. Cela fait suite notamment aux importants écarts-types sur nos trois métriques sur notre exemple jouet sans-fil. Nous verrons aussi une première évaluation de Q-routing sur un scénario avec mobilité. En effet, nous pensons que la mobilité aura un rôle accru dans les ateliers de l'industrie du futur.

Chapitre 7

Amélioration de la mesure du délai de Q-routing

7.1 Introduction

Nous avons vu dans le précédent chapitre comment atténuer le comportement glouton de Q-routing. Q-routing a obtenu de bons résultats sur la grille adaptée de Boyan et Littman. Néanmoins, les résultats sur l'exemple jouet sans-fil sont décevants. D'autant que, ce scénario était censé avantager Q-routing. Notre hypothèse est que le délai est plus difficile à estimer sur les réseaux sans-fil. Dans ce chapitre, nous allons vérifier cette hypothèse. Nous proposons aussi une première évaluation de notre implémentation sur un scénario avec mobilité. L'objectif final est d'améliorer l'estimation des Q-valeurs. En effet, l'estimation du délai peut être difficile notamment avec les topologies sans-fil. Nous allons montrer sur un simple cas de mobilité que la valeur des Q-valeurs peut être bruitée. Nous en déduisons que la mesure du délai n'est pas correcte. En effet, les Q-valeurs sont calculées à partir de cette mesure. Q-routing fait donc de mauvais choix de routage car ces Q-valeurs sont sa métrique de routage. Ce cas se produit surtout lorsque deux routes ont des Q-valeurs proches comme nous allons le voir dans le prochain scénario. Nous proposons d'utiliser une méthode de filtrage pour améliorer l'estimation des Q-valeurs.

Le plan de ce chapitre est le suivant, nous mettons en évidence les problèmes d'estimations des Q-valeurs sur un scénario avec mobilité (Section 7.2). Nous détaillons ensuite notre proposition (Section 7.3). Nous l'évaluons ensuite sur notre scénario avec mobilité ainsi que sur la topologie sans-fil adaptée de la grille de Boyan et Littman (Section 7.4). Enfin, nous terminons par une conclusion.

7.2 Mise en évidence des problèmes d'estimation

7.2.1 Présentation du problème

La Q-valeur est calculée à partir de la mesure du délai. Il s'agit des paramètres $q + s$ dans l'Equation 4.3. Les Q-valeurs sont indirectement sensibles aux éventuelles erreurs de mesure. Sauf qu'elles sont finalement les métriques de routage de Q-routing donc Q-routing peut faire des mauvais choix. Ce problème a été découvert lors de l'évaluation de Q-routing sur notre scénario avec mobilité.

7.2.2 Scénario

Notre scénario est constitué d'une topologie de 6 nœuds. Les nœuds communiquent via des liens IEEE 802.11a 9 Mb/s. La couche liaison est de type IEEE 802.11e, c'est-à-dire la couche liaison la plus récente disponible. Le nœud 9 commence à envoyer des données au nœud 10 au bout de 30 s de temps simulé. Il s'arrête une minute avant la fin de la simulation. Le débit est de 1 Mb/s ainsi le lien radio n'est pas saturé. Le nœud 9 se déplace d'Est en Ouest à partir de la 35e seconde et s'arrête 1 minute avant la fin de la simulation. Il parcourt une distance de 1300 m au total à une vitesse uniforme au cours de la simulation. La durée simulée dépend de la vitesse du nœud 9. La Figure 7.1 illustre ce scénario. Afin de rendre Q-routing plus réactif au changement de topologie, l'intervalle entre chaque mise à jour périodique est de 5 s (au lieu de 10 s). Le délai avant péremption des routes est aussi abaissé à 10 s (au lieu de 60 s). Le Tableau 7.1 récapitule les paramètres de simulation. Les paramètres non précisés sont laissés par défaut.

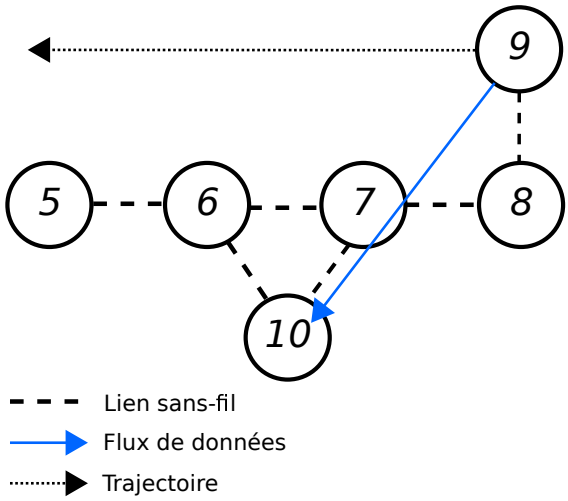


FIG. 7.1 : Scénario avec mobilité utilisé pour mettre en évidence le problème d'estimation du délai.

Élément	Paramètre	Valeur
Simulation	Graines	30 graines différentes
Réseau	Lien	IEEE 802.11a 9 Mb/s couche liaison IEEE 802.11e
Nœud	File d'attente	1 file FIFO
Données applicatives	Taille par envoi Début Fin	512 octets 30 s 1 min avant la fin de simulation
Q-routing	η Phase de découverte	0,9 15 s

TAB. 7.1 : Paramètres de simulation pour notre scénario avec mobilité.

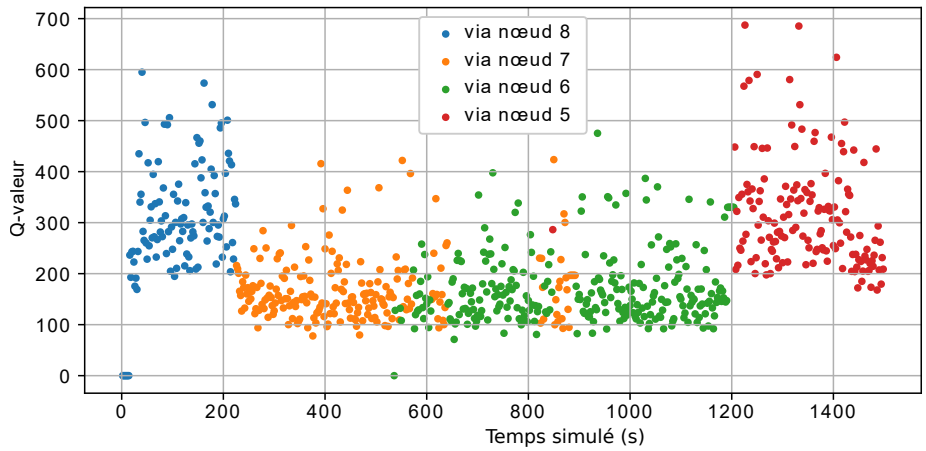


FIG. 7.2 : Q-valeurs en fonction du temps simulé sur notre scénario avec mobilité.

7.2.3 Résultats

Dans cette section, nous nous focaliserons uniquement sur les Q-valeurs et sur le nombre de changements de routes. Les valeurs du taux de paquets reçus et le délai de livraison ne sont pas pertinentes dans l'illustration du problème. Les résultats ont été obtenus avec Qualnet 9.2.1. La Figure 7.2 représente les Q-valeurs des routes actives du nœud 9 et à destination du nœud 10 en fonction du temps simulé pour la graine 7 pour une vitesse de 1 m/s. Les Q-valeurs sont relevées toutes les deux secondes. Le nœud 9 commence par utiliser la route via le nœud 8, il s'agit de la seule route disponible. Les premières Q-valeurs de la route via 8 sont 0 à cause de l'initialisation optimiste. Les autres Q-valeurs pour cette route sont comprises entre 150 et 600. À partir de 220 s, le nœud 9 utilise la route via le nœud 7. Les Q-valeurs de la route via 7 sont comprises entre 80 et 440. À 520 s, la route via le nœud 6 est choisie car elle a la plus petite Q-valeur à cause de l'initialisation optimiste. Entre 520 s et 650 s, le nœud 9 utilise alternativement la route via 7 et celle via 6. Cette

situation se produit aussi entre 800 s et 900 s. À 850 s, la route via le nœud 5 est momentanément utilisée. Les Q-valeurs de la route via 6 sont comprises entre 80 et 500. Entre 1200 s et la fin de la simulation, le nœud 9 utilise la route via le nœud 5. Ses Q-valeurs sont comprises entre 150 et 700.

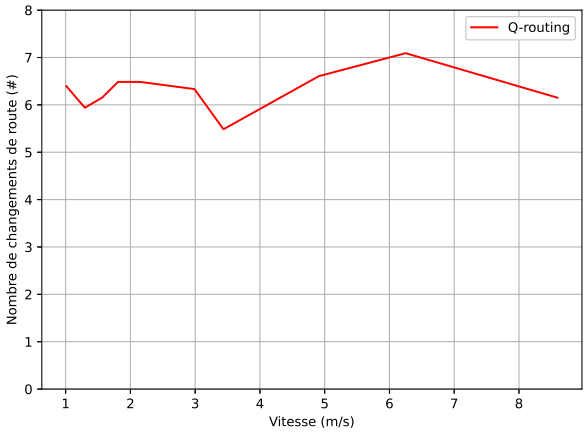


FIG. 7.3 : Nombre de changements de route en fonction de la vitesse pour Q-routing.

Les routes via 7 et 6 ont le même nombre de sauts. Comme les liens sans-fil ne sont pas saturés, il est normal que leurs Q-valeurs soient proches. Malheureusement, la moindre erreur de mesure peut entraîner un changement de route. En effet, il y a 11 changements de routes dans cette simulation par exemple. Nous avons vérifié si cette situation se produit pour d'autres graines et pour d'autres vitesses. Pour cela, nous utilisons le nombre moyen de changements de routes. La Figure 7.3 représente ce nombre moyen de changements de route en fonction de la vitesse. Le nombre de changements de route est compris entre 5,5 et 7,5 changements. Ce résultat montre que le phénomène se produit pour d'autres graines et pour d'autres vitesses mais dans des proportions moindres. Le nombre de changements reste élevé. Par exemple, il est seulement de 3 pour nuOLSRv2 indépendamment de la vitesse. Ces changements de routes intempestifs favorisent notamment le désordonnancement des paquets car deux paquets consécutifs peuvent passer par deux routes différentes.

L'estimation du délai n'est pas la seule cause des changements de route intempestifs. En effet, l'initialisation optimiste pose problème dans le cas d'un scénario avec mobilité. L'utilisation éphémère de la route via 5 à 850 s le montre. Le nœud 9 commence à être à portée du nœud 5. Cependant, celui-ci est pour le moment plus éloigné que les nœuds 6 et 7. La route via 5 ne devrait pas être sélectionnée à ce moment. Sauf que la Q-valeur associée est de 0 à cause de l'initialisation optimiste. Le problème n'apparait pas dans le cas de topologie statique, car les distances entre les nœuds ne changent pas en cours du temps.

7.3 Solutions proposées

Dans la section précédente, nous avons mis en évidence deux problèmes. Le premier concerne l'estimation du délai. En effet, la mesure du délai est bruitée. Nous proposons d'utiliser une méthode de filtrage afin de résoudre ce problème. Le second problème mis en avant est lié à l'initialisation optimiste dans les scénarios avec mobilité. Nous proposons un système de trois statuts afin de s'assurer de la fiabilité de la route.

7.3.1 Filtrage des mesures de délais

Notre problème de mesure est un problème de traitement du signal. Nous proposons d'utiliser une méthode de filtrage. Il existe de nombreuses méthodes de filtrage comme les filtres de Kalman [Li+15] et de Savitzky-Golay [Sch11]. Chaque méthode de filtrage a son efficacité et sa complexité. Nous ne souhaitons pas augmenter la complexité de Q-routing donc nous avons opté pour la moyenne glissante. La moyenne glissante est une méthode de filtrage simple à implémenter, peu complexe et paramétrable. Nous avons choisi de garder un nombre fixe de mesures pour effectuer le filtrage. Ces mesures peuvent avoir un espacement temporel différent entre elles. Afin de conserver une cohérence entre les mesures, celles-ci sont supprimées si la route se périme. Un inconvénient est que de l'espace mémoire supplémentaire est nécessaire en fonction du nombre de mesures conservées.

7.3.2 Route à trois statuts

L'initialisation optimiste pose problème en scénario de mobilité. En effet, les nouvelles routes peuvent être sélectionnées prématurément car leur Q-valeur est initialisée à 0. Dans notre implémentation originale, une route ne peut avoir que deux statuts : valide et périmé. Nous proposons d'étendre à trois statuts afin d'éviter la sélection prématurée d'une route. Le premier est le statut « en probation ». Il est attribué aux nouvelles routes ainsi qu'aux routes périmées recevant une mise à jour. Le second est le statut « vérifié ». Il est attribué dès qu'une route reçoit au moins 3 mises à jour. Enfin, nous conservons le statut « périmé » pour les routes n'ayant pas reçu de mises à jour pendant une durée déterminée. Le choix des routes a aussi été modifié pour prendre en compte ces modifications. Q-routing va d'abord choisir parmi les routes vérifiées disponibles. S'il n'y a pas de route vérifiée disponible, alors une route « en probation » sera choisie.

7.4 Évaluation des solutions proposées

Dans cette section, nous proposons d'évaluer les propositions précédentes dans divers scénarios. L'ensemble des résultats a été obtenus avec Qualnet 9.2.1.

7.4.1 Scénario avec mobilité

Dans un premier temps, nous allons réutiliser le scénario de la sous-section 7.2.2 (Figure 7.1). Cette fois, l'objectif est d'évaluer Q-routing dans un scénario de mobilité. Dans un premier temps, nous allons évaluer Q-routing avec filtrage. Le système de route à trois statuts est présent sur notre version modifiée de Q-routing. Nous nous intéresserons d'abord aux Q-valeurs. Ensuite, nous comparerons Q-routing et sa version modifiée grâce aux métriques de qualités de services habituelles (taux de paquets reçus, délai de livraison et gigue). Nous utiliserons nuOLSRv2 comme protocole de routage de référence. Cela nous permettra d'obtenir une première évaluation de notre implémentation de Q-routing sur un scénario avec mobilité.

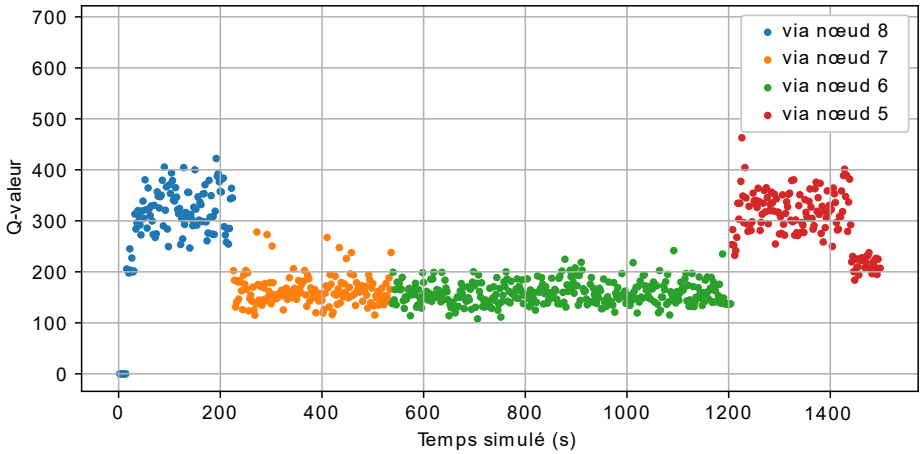


FIG. 7.4 : Q-valeurs en fonction du temps simulé sur notre scénario avec mobilité avec Q-routing avec filtrage.

La Figure 7.4 représente les Q-valeurs des routes actives du nœud 9 et à destination du nœud 10 en fonction du temps simulé pour la graine 7 pour une vitesse de 1 m/s pour Q-routing avec filtrage. Le nœud 9 commence par utiliser la route via le nœud 8 car il s'agit de la seule route disponible. Les premières Q-valeurs de la route via 8 sont 0 à cause de l'initialisation optimiste toujours présente. Les Q-valeurs sont comprises en 190 et 420. À partir de 220 s, le nœud 9 choisit la route via 7. Les Q-valeurs sont comprises entre 110 et 290. La route via 7 est utilisée sans interruption jusqu'à la 550e seconde. La route via le nœud 6 est ensuite utilisée. Ces Q-valeurs sont comprises entre 100 et 250. La route via le nœud 5 est enfin utilisée à partir de 1200 s et jusqu'à la fin de la simulation. Ces Q-valeurs sont comprises entre 180 et 480. Il y a seulement trois changements de routes.

Pour confirmer ces observations, nous regardons le nombre de changements de routes en fonction de la vitesse représenté par la Figure 7.5. Le nombre de changements de route est compris entre 5,5 et 7 pour Q-routing. Q-routing avec filtrage change donc moins de route par rapport à sa version originale. En moyenne, il ne dépasse pas les 4 changements.

La Figure 7.6 représente le taux de paquets reçus en fonction de la vitesse pour Q-routing, sa modification et nuOLSRv2. Nous rappelons que les paquets désordonnés sont détruits par la

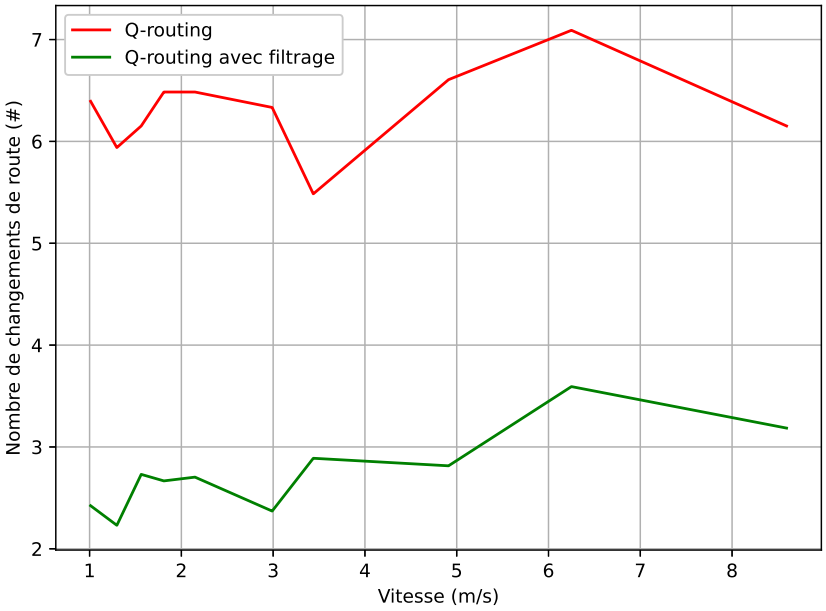


FIG. 7.5 : Nombre de changements de route en fonction de la vitesse pour les deux versions de Q-routing.

destination. nuOLSRv2 offre le meilleur taux de paquets reçus. À 1 m/s, nuOLSRv2 transmet 99 % des paquets. A 9 m/s, le taux de paquets reçus est 94 %. Q-routing livre 98 % des paquets à 1 m/s. Le taux de paquets livrés de Q-routing diminue plus rapidement qu'avec nuOLSRv2. À 9 m/s, Q-routing ne livre plus que 85 %. Q-routing avec filtrage a le même profil de courbe que Q-routing. Il livre 98 % des paquets à 1 m/s. À partir 3,5 m/s, Q-routing avec filtrage livre un peu plus de paquets que sa version originale. À 9 m/s, la différence n'est que de 1 %.

La Figure 7.7 représente le délai moyen de livraison en fonction de la vitesse pour Q-routing, sa modification et nuOLSRv2. nuOLSRv2, notre référence ici, livre les paquets le plus rapidement indépendamment de la vitesse. À 1 m/s, il livre les paquets en 44 ms. À 9 m/s, il faut 289 ms pour que les paquets atteignent leur destination. Q-routing met 62 m/s à 1 ms pour acheminer les paquets. À vitesse maximale, le délai de livraison passe à 493 ms, soit plus de 200 ms de plus que nuOLSRv2. Q-routing avec filtrage est un peu plus rapide que Q-routing. À basse vitesse, les paquets sont acheminés en 57 ms. Au maximum, ils sont livrés en 472 ms, soit 20 ms de moins que la version originale.

Enfin, nous nous intéressons à la gigue. La Figure 7.8 représente la gigue moyenne en fonction de la vitesse pour Q-routing, sa version modifiée et nuOLSRv2. nuOLSRv2 offre la plus petite gigue indépendamment de la vitesse. Q-routing et Q-routing avec filtrage ont une gigue similaire entre 1 m/s et 3,5 m/s. À partir de 3,5 m/s, la variation du délai est un peu plus importante avec Q-routing qu'avec sa version avec filtrage. Elle est de 0,9 ms à 9 m/s pour Q-routing et 0,8 ms pour Q-routing

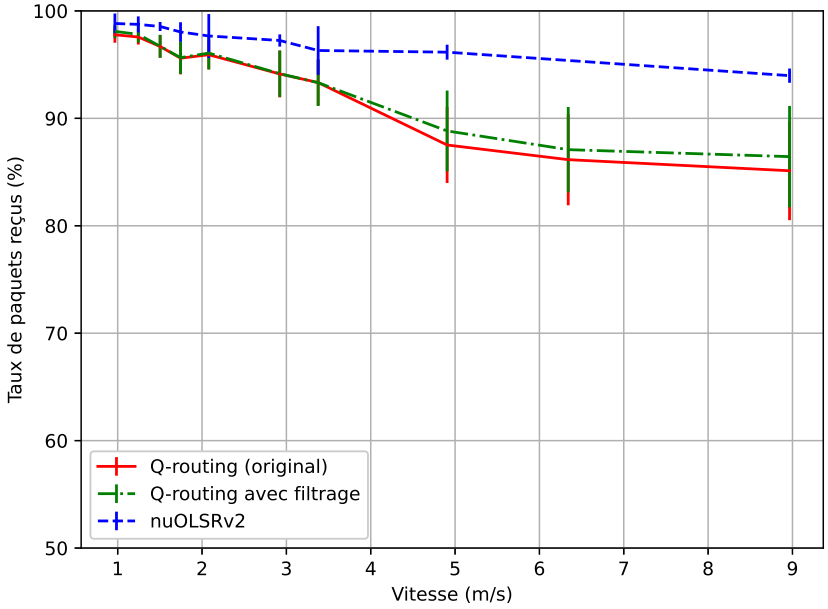


FIG. 7.6 : Taux de paquets reçus en fonction de la vitesse du nœud 9

avec filtrage.

Conclusions

Dans ce premier scénario, nous avons évalué Q-routing sur un cas simple de mobilité. Jusqu'à 3,5 m/s, les résultats de Q-routing sont acceptables en comparaison avec celle de nuOLSRv2 qui a été spécialement conçu pour des changements fréquents de topologie. Au-delà de 3,5 m/s, l'écart entre Q-routing et nuOLSRv2 se creuse pour toutes les principales métriques de qualité de service. Notre implémentation de Q-routing est minimale. Il n'y a aucune fonction auxiliaire contrairement à nuOLSRv2. Par exemple, notre implémentation ne comporte pas de table de voisinage. Or, cette fonctionnalité est importante dans les scénarios comportant de la mobilité. Elle permet notamment d'anticiper la disparition d'une route.

À partir de l'évaluation de nos modifications, nous en déduisons que le filtrage est la solution la plus efficace pour améliorer l'estimation du délai. Le filtrage permet de diminuer la dispersion des Q-valeurs. Cela à pour effet de diminuer le nombre de changements de route. En effet, dans notre cas, les routes via 6 et via 7 ont des Q-valeurs proches car les deux routes ont la même distance. La moindre erreur peut entraîner un changement de route. En plus de diminuer le nombre de changements de route, Q-routing avec filtrage livre plus de paquets et plus rapidement que Q-routing à partir de 3,5 m/s. Le gain reste toutefois très modeste.

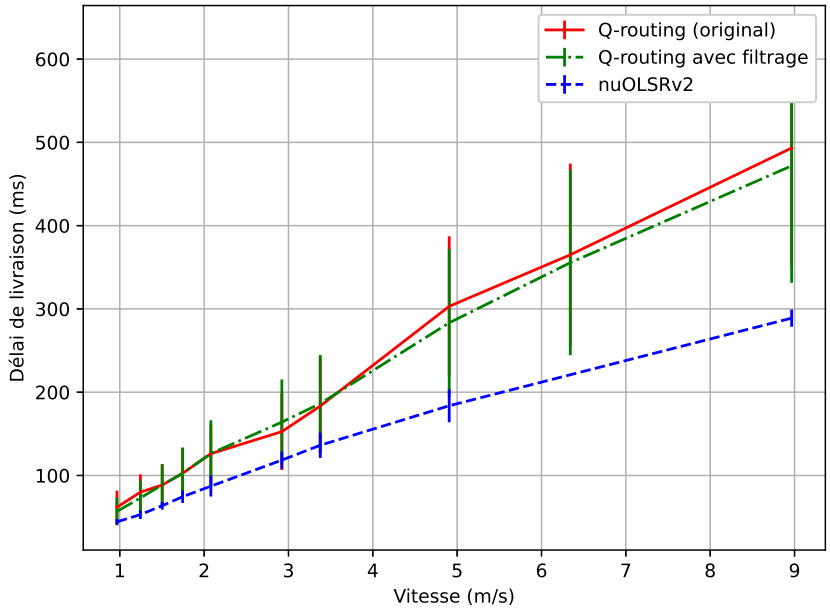


FIG. 7.7 : Délai de livraison moyen en fonction de la vitesse du nœud 9

7.4.2 Grille sans-fil inspirée de Boyan et Littman

Nous allons maintenant évaluer Q-routing avec épisodes sur une topologie plus grande : la version sans-fil et adaptée de la grille de Boyan et Littman. La Figure 6.8 illustre cette topologie. Dans ce scénario, les couples sources - destinations seront répartis aléatoirement sur la grille et dans le temps. Le but est de vérifier les apports en termes de qualité de services du filtrage sur une grande topologie sans-fil et avec un trafic diffus. Le Tableau 6.2 récapitule les paramètres du scénario. Pour les deux scénarios, nous évaluons Q-routing, Q-routing avec filtrage et nuOLSRv2.

La Figure 7.9 illustre le taux de paquets livrés en fonction du débit d'arrière-plan pour Q-routing et sa version modifiée ainsi que pour nuOLSRv2. Q-routing et Q-routing avec filtrage ont le même profil de courbe. En moyenne, Q-routing avec filtrage offre le meilleur taux de livraison. La différence entre Q-routing et sa version modifiée est très réduite. Au mieux, Q-routing avec filtrage livre seulement 1 % de paquets de plus à 800 kb/s par source de trafic. L'écart avec nuOLSRv2 est de 6 % à 800 kb/s par source. L'écart entre Q-routing et nuOLSRv2 est donc de 5 % à 8 %.

La Figure 7.10 illustre le taux de paquets livrés en fonction du débit d'arrière-plan. Q-routing et Q-routing avec filtrage ont le même profil de courbe. En moyenne, Q-routing avec filtrage offre le meilleur délai de livraison. Les deux versions de Q-routing offrent un délai de livraison inférieur à celui de nuOLSRv2. Le délai de livraison augmente pour tous les protocoles entre 0 et 1,7 Mb/s par source. C'est entre 400 kb/s et 800 kb/s que l'augmentation est la plus forte pour les trois protocoles. À partir 1,7 Mb/s par source, le délai de livraison augmente moins vite rapidement avec nuOLSRv2.

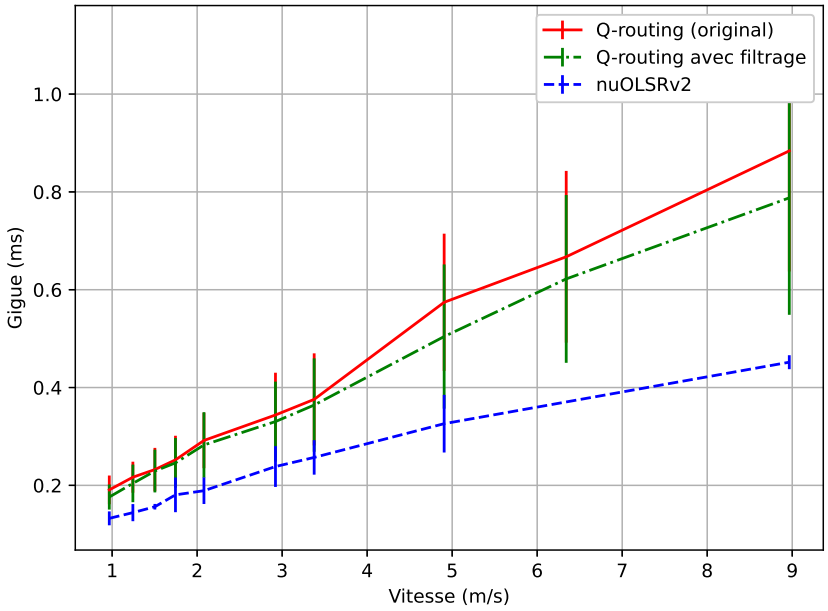


FIG. 7.8 : Gigue moyenne en fonction de la vitesse du nœud 9

Il est déjà de 9 s. Le délai de livraison stagne à 8,5 s pour Q-routing sur cet intervalle. Pour Q-routing avec filtrage, le délai de livraison augmente jusqu'à dépasser celui de Q-routing à 2,1 Mb/s. À partir 2,1 Mb/s, après avoir stagné, le délai augmente avec Q-routing pour atteindre 8,8 s. Le délai de livraison de Q-routing avec filtrage diminue jusqu'à 8,3 s. Il livre donc 0,5 s plus rapidement que Q-routing et 0,8 s plus rapidement que nuOLSRv2 lorsque la charge réseau est élevée.

Enfin, la Figure 7.11 représente la gigue moyenne en fonction du débit par source. nuOLSRv2 a la gigue moyenne la plus élevée. Q-routing et Q-routing avec filtrage ont le même profil de courbe. La gigue se dégrade plus rapidement avec nuOLSRv2 qu'avec les deux versions de Q-routing. L'écart est de 5 ms à 800 kb/s par source entre Q-routing et nuOLSRv2. Il est de 10 ms à 1,7 Mb/s.

Conclusions

Dans ce second scénario, nous avons évalué Q-routing avec filtrage sur une topologie complexe mais sans mobilité. Q-routing avec filtrage obtient de meilleurs résultats que ceux de nuOLSRv2 pour les trois métriques évaluées. Le taux de paquets reçus est similaire entre les deux versions de Q-routing. En termes de délai de livraison, Q-routing avec filtrage offre une amélioration par rapport à Q-routing avec un gain d'une demie seconde à partir de 1,7 Mb/s. Ce gain est d'autant plus intéressant qu'il ne s'accompagne pas de perte de paquets. Concernant la gigue moyenne, Q-routing avec filtrage apporte en moyenne un très léger bénéfice par rapport à sa version originale.

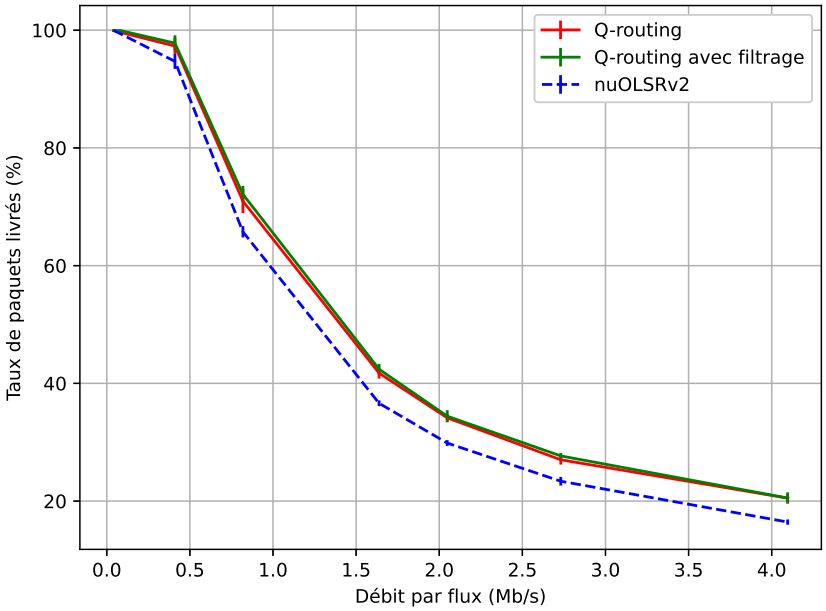


FIG. 7.9 : Taux de paquets reçus pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil.

7.5 Conclusions

Dans ce chapitre, nous avons mis en évidence que les Q-valeurs étaient bruitées. Nous en déduisons que la mesure du délai est aussi bruitée. Ce problème conduit Q-routing à prendre de mauvaises décisions de routage et à changer trop fréquemment de routes. Nous avons aussi constaté que l'initialisation optimiste des Q-valeurs est inadaptée à la mobilité. En effet, celle-ci force l'utilisation des nouvelles routes prématurément. Afin de résoudre ces deux problèmes, nous avons proposé deux modifications : *i)* l'enrichissement des statuts des routes, *ii)* l'utilisation du filtrage. Nos simulations montrent que la combinaison du système de trois statuts des routes et de l'utilisation du filtrage permet de diminuer efficacement le nombre de changements de route. Ils améliorent aussi légèrement le taux de paquets reçus. Les performances de Q-routing restent néanmoins inférieures comparées à celles de nuOLSRv2. Cela peut notamment s'expliquer par les fonctions auxiliaires de nuOLSRv2. Sur la grille sans-fil adaptée de Boyan et Littman, Q-routing avec filtrage permet une amélioration du délai de livraison tout en livrant autant de paquets.

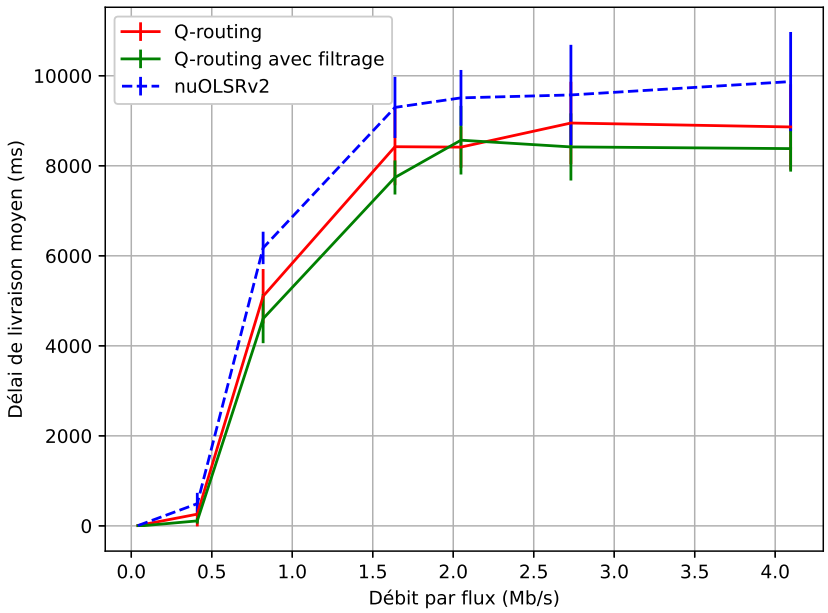


FIG. 7.10 : Délai de livraison moyen pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil.

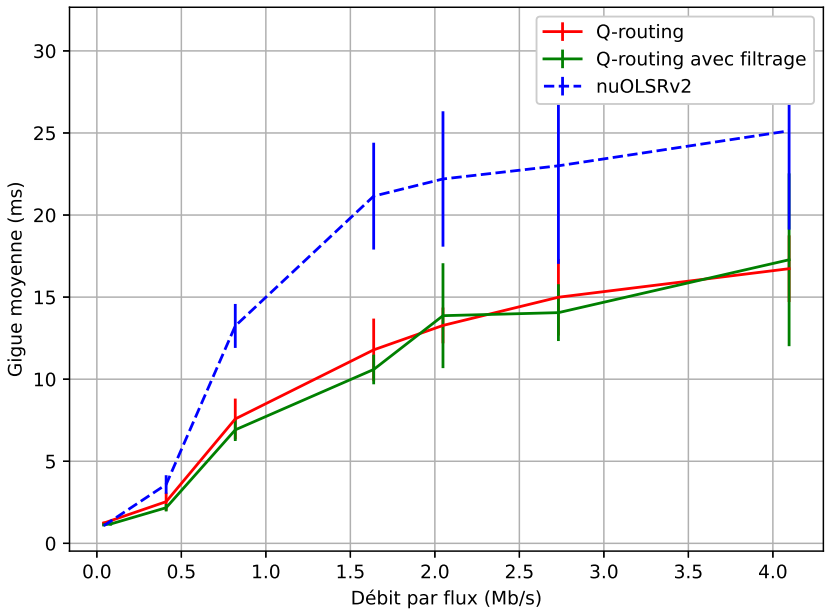


FIG. 7.11 : Gigue moyenne pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil.

Conclusion de la deuxième partie

Dans le Chapitre 5, nous avons détaillé notre implémentation de Q-routing. Nous l'avons évalué sur deux topologies filaires incluant la grille irrégulière de Boyan et Littman. OLSRv2 (RFC 7181) était notre protocole de routage de référence, car c'est un protocole de routage standardisé et spécialisé dans le routage ad-hoc. Les résultats sont bons confirmant la pertinence de nos choix d'implémentation.

Dans le Chapitre 6, nous avons mis en évidence un effet de la stratégie gloutonne de Q-routing. En effet, Q-routing est sensible aux optimums locaux. Comme nous l'avons montré, une congestion momentanée peut priver Q-routing d'une route. Nous avons proposé deux solutions inspirées de l'apprentissage automatique pour atténuer cette particularité : *i)* les mises à jour ε-gloutonne et *ii)* le système d'époques. Le système d'époques s'est avéré le plus efficace. Nous l'avons comparé à la version originale de Q-routing et OLSRv2 sur deux topologies sans-fil. Les résultats de Q-routing avec époques sont meilleurs que ceux de Q-routing sur la grille sans-fil.

Enfin, dans le Chapitre 7, nous nous sommes concentrés sur l'amélioration de la métrique de routage. En effet, Q-routing mesure le délai afin de calculer sa métrique de routage. Nous avons montré que cette métrique pouvait être bruitée car la mesure de délai n'est pas parfaite. Ceci conduit Q-routing à effectuer de mauvais choix de routage. Nous avons donc proposé une solution pour améliorer l'estimation du délai. Le filtrage via une moyenne glissante est efficace et permet de diviser par deux le nombre de changements de route. De plus, dans le scénario utilisant la grille sans-fil adaptée de Boyan et Littman, Q-routing avec filtrage permet une amélioration des métriques de qualité de services (taux de paquets reçus, délai de livraison et gigue). Notre implémentation a aussi montré ses limites sur notre scénario avec mobilité. En effet, contrairement à OLSRv2, notre implémentation ne dispose pas de fonctions auxiliaires comme une table de voisinage par exemple.

Conclusion générale

Dans cette thèse, nous avons commencé par présenter un état de l'art dans la première partie en lien avec l'Industrie 4.0 et les réseaux cognitifs. Dans le Chapitre 1, nous avons présenté l'Industrie 4.0. Elle est évoquée pour la première fois en 2011 en Allemagne, puis formalisé deux ans plus tard par un groupe d'expert [Kag13]. Son objectif principal est l'augmentation de la compétitivité sans augmenter pour autant les volumes de production. Il est donc nécessaire d'améliorer la gestion des diverses ressources de production, que ce soit les matières premières ou l'énergie. Il faut aussi augmenter la flexibilité des processus de fabrication. Cela nécessite que l'usine soit reconfigurable [Gri17]. Pour cela, l'Industrie 4.0 s'appuie notamment sur l'Internet des Objets et les systèmes cyber-physiques.

Pour supporter les contraintes techniques de l'Industrie 4.0 comme la reconfiguration de l'usine et l'Internet des objets, les réseaux sans-fil sont la principale solution. Dans le Chapitre 2, nous avons abordé les technologies des réseaux industriels. Avant les années 2000, ces technologies étaient surtout des bus de terrain propriétaires incompatibles avec les réseaux informatiques. Avec l'augmentation des réseaux informatiques à la fin des années 90, de nouvelles technologies sont apparues. Ces nouvelles technologies restent propriétaires, mais elles s'appuient sur des standards IEEE comme IEEE 802.3 et IEEE 802.15.4. À partir de 2010, ces deux standards IEEE ont commencé à intégrer des fonctionnalités présentes dans les technologies des réseaux industriels notamment à travers Ethernet TSN et IEEE 802.15.4 TSCH. Parallèlement, les réseaux cellulaires ont aussi évolué pour supporter des scénarios liés à l'industrie. Par exemple, l'Internet des Objets industriel est un des cas d'usage de la 5G au travers notamment du protocole NB-IoT. Cela montre une réelle convergence des technologies des réseaux informatiques et des réseaux industriels (*IT*/*OT*).

Dans le Chapitre 3, nous évoquons notamment le risque d'interférences lié à la multiplication des réseaux sans-fil. En effet, la plupart des technologies sans-fil utilisent les mêmes bandes de fréquences, notamment les bandes ISM 868 MHz et 2,4 GHz. Cette surutilisation des bandes ISM augmente le risque d'interférences donc de perte de données. Par exemple, lorsque IEEE 802.11 et IEEE 802.15.4 entrent en concurrence, jusqu'à 80 % des trames IEEE 802.15.4 peuvent être détruites. Nous avons relevé quelques solutions pour limiter les risques d'interférences. Néanmoins, il n'est pas toujours possible de limiter ces interférences, notamment à cause des contraintes technologiques ou la présence d'un environnement défavorable (présence de surfaces réfléchissantes par exemple). Nous avons présenté deux autres moyens d'augmenter les performances d'un réseau : *i)* l'optimisation inter-couches et *ii)* les réseaux cognitifs. Les optimisations inter-couches reposent

sur la communication et l'interaction entre des couches de services potentiellement non adjacentes pour améliorer les performances. Mais, les optimisations inter-couches ne respectent pas les principes de la modélisation OSI, ce qui impacte grandement leur pérennité. Les réseaux cognitifs sont une généralisation de la radio cognitive à l'ensemble des couches de la modélisation OSI. Les réseaux cognitifs peuvent aussi intégrer une ou plusieurs optimisations inter-couches. Ils peuvent donc avoir un ou plusieurs objectifs comme la maximisation du débit, la minimisation de la latence ou l'amélioration de la fiabilité. Bien que le concept de réseaux cognitifs ait été exploré, il n'existe que très peu d'implémentations. Les réseaux cognitifs cumulent deux inconvénients : leur faible pérennité et leur grande complexité. Face à cette complexité, la plupart des auteurs ont préféré se concentrer sur l'amélioration d'une fonctionnalité ou d'une couche protocolaire.

L'amélioration d'une fonctionnalité ou d'une couche protocolaire peut s'effectuer par différents moyens. Dans ce travail, nous avons choisi d'utiliser l'apprentissage par renforcement. Dans le Chapitre 4, nous avons commencé par présenter le concept d'apprentissage par renforcement. L'apprentissage par renforcement est un des trois paradigmes de l'apprentissage automatique. L'apprentissage s'effectue par essais et erreurs et ne nécessite pas nécessairement de jeu de données initial. Cela convient à l'environnement dynamique que sont les réseaux informatiques. Nous avons présenté plusieurs algorithmes de la famille de Q-learning, puis quelques applications. Nous nous sommes focalisés sur Q-routing. Q-routing est un algorithme de routage inspiré de Q-learning, un algorithme d'apprentissage par renforcement. Bien que les résultats obtenus par Boyan et Littman soient intéressants, ils sont difficilement reproductibles par le simulateur utilisé. Il existe plusieurs dizaines de travaux d'améliorations de Q-routing. Une partie de ces travaux se concentre sur l'amélioration de l'algorithme. L'autre partie se focalise sur des scénarios qui n'existaient pas à la publication de l'article de Boyan et Littman comme les réseaux de capteurs ou la radio cognitive.

Dans la seconde partie, nous avons présenté nos contributions. Dans le Chapitre 5, nous avons décrit notre implémentation de Q-routing. Nous avons spécifié les éléments propres à notre implémentation comme les mises à jour et la mesure du délai. Notre implémentation a été réalisée sur le simulateur réseau Qualnet. Afin de vérifier la qualité de notre implémentation, nous l'avons évalué sur deux topologies filaires. Le premier scénario nous a servi à mettre en avant la capacité de Q-routing à utiliser un chemin alternatif en cas de congestion. Le second scénario utilise la grille irrégulière originale de Boyan et Littman. Son objectif est de reproduire les résultats présentés dans leur article. Nous avons utilisé le protocole de routage standardisé et spécialisé OLSRv2 (RFC 7181) comme protocole de référence. D'après nos simulations, Q-routing est capable d'utiliser un chemin alternatif contrairement à OLSRv2. Sur les deux scénarios, Q-routing offre un meilleur taux de paquets reçus et un meilleur délai de livraison.

Dans le Chapitre 6, nous nous sommes focalisés sur un effet de la stratégie gloutonne de Q-routing. Nous avons commencé par mettre en évidence que Q-routing était sensible aux optimums locaux. Dans les faits, une congestion momentanée peut priver Q-routing d'une route. Nous avons proposé deux solutions inspirées de l'apprentissage automatique pour atténuer ce phénomène. La première solution s'inspire des stratégies ε-gloutonnes. Il s'agit des mises à jour ε-gloutonnes. La deuxième solution est proche conceptuellement de la solution retenue par Choi et Yeung [CY96], il s'agit d'une réinitialisation périodique des Q-valeurs. Nous avons évalué nos deux propositions

sur un scénario avec une topologie sans-fil. D'après nos simulations sur ce scénario, la réinitialisation périodique est plus efficace, car elle ne dépend pas d'une approche probabilité. Nous avons donc évalué Q-routing et sa version modifiée sur deux topologies sans-fil dont une version adaptée de la grille irrégulière de Boyan et Littman. Sur la première topologie, les résultats de notre version modifiée de Q-routing sont moins bons que ceux de sa version originale. Nous supposons que les paramètres introduits par notre modification influencent les performances. Sur la version adaptée de la grille irrégulière, les résultats obtenus par notre version modifiée sont supérieurs à ceux Q-routing sur les trois métriques proposées (taux de paquets reçus, délai de livraison et gigue). Néanmoins, selon le débit et le scénario, ces résultats sont parfois inférieurs à ceux d'OLSRv2.

Enfin, dans le Chapitre 7, nous nous sommes concentrés sur l'amélioration de la métrique de routage. Q-routing utilise notamment le délai afin de calculer les Q-valeurs. Dans un scénario avec mobilité, nous avons mis en évidence à travers les Q-valeurs que le délai était difficile à estimer. Dans notre scénario, cela se traduit par des changements de routes intempestifs. L'initialisation optimiste contribue aussi à ce phénomène en forçant l'utilisation des nouvelles routes. Nous avons proposé de remplacer l'initialisation optimiste par un système à trois statuts : en probation, vérifié et périmé. Le passage d'un statut à l'autre est conditionné par les mises à jour. En complément, nous avons proposé une approche s'appuyant sur le filtrage du délai. Dans cette approche, nous avons appliqué une méthode de filtrage sur la valeur mesurée. Nous avons évalué ces deux approches sur notre scénario avec mobilité. Le filtrage permet de diviser le nombre de changements de route par deux. Le gain en termes de taux de paquets reçus et de délai dépend des paramètres du scénario. Sur la version sans-fil de la grille irrégulière, le filtrage permet d'améliorer le délai de livraison en maintenant le taux de paquets reçus.

Q-routing offre en moyenne les meilleures performances en termes de taux de paquets reçus et de délai de livraison dans 4 des 6 scénarios d'évaluation présentés. Deux points restent néanmoins à améliorer sur notre implémentation. D'un part, la variation du délai avec Q-routing est aussi plus élevée qu'avec OLSRv2 dans 4 des 6 scénarios. Cette métrique est aussi importante que le taux de paquets reçus et le délai de livraison dans les réseaux industriels. D'autre part, les résultats de Q-routing sur notre scénario avec mobilité sont moins bons que ceux d'OLSRv2. Cela peut s'expliquer notamment par la différence de maturité entre les deux protocoles. Des efforts supplémentaires en termes de conception protocolaire sont donc nécessaires afin que Q-routing puisse devenir le protocole de routage des réseaux de l'Industrie 4.0.

Pour rappel, nos six contributions classées par ordre d'apparition dans ce document sont :

- l'implémentation de Q-routing sur le simulateur réseau Qualnet ;
- l'évaluation de notre implémentation sur deux topologies filaires dont la grille irrégulière de Boyan et Littman ;
- la proposition de deux approches qui servent à atténuer un des effets de la stratégie gloutonne de Q-routing ;
- l'évaluation de Q-routing sur deux topologies sans-fil dont une version adaptée de la grille irrégulière de Boyan et Littman ;

- la proposition d'une approche pour améliorer l'estimation de la métrique de routage par le filtrage ;
- l'évaluation de Q-routing dans un scénario comportant de la mobilité.

Perspectives

D'autres aspects n'ont pas pu être traités lors de cette thèse. Les perspectives concernent à la fois l'implémentation de Q-routing ainsi que diverses améliorations possibles comme l'ajout d'autres métriques dans le calcul des Q-valeurs.

Métrique de routage multi-objectifs

Nous avons constaté dans les Chapitres 5 et 7 que le délai était difficile à mesurer et que cette estimation peut être bruitée. D'autres métriques pourraient aider Q-routing à effectuer son choix de routage. Par exemple, le nombre de trames détruites ou de retransmission permettrait d'estimer la capacité effective du lien. Les retransmissions sont de plus une cause possible de l'augmentation du délai. D'autres métriques comme le RSS (*Received Signal Strength*) permettraient d'estimer la distance entre chaque routeur. Dans les scénarios avec mobilité, cette métrique aiderait à anticiper des ruptures de liens sans-fil par exemple. Ces métriques seraient utilisées dans le calcul des Q-valeurs.

Double Q-routing

Dans le Chapitre 4, nous avons abordé quelques algorithmes d'apprentissage par renforcement notamment Q-learning et ses dérivés. Néanmoins, Q-learning n'est pas parfait. Il a tendance à surestimer les récompenses. Cela se traduit par une prise de risque accrue. Heureusement, ce problème a été corrigé dans Double Q-learning [Has10]. La principale différence réside dans la présence de deux Q-fonctions et leurs mises à jour. Q-routing s'inspire de Q-learning. Il est donc envisageable de modifier Q-routing afin qu'il reprenne les améliorations de Double Q-learning. Le coût en mémoire est certes plus élevé que celui de Q-routing, mais il équivaut finalement à celui de Q-routing avec époques (Section 6.3.2).

Implémentation et expérimentation sur matériels

Nous avons implémenté Q-routing sur le simulateur réseau Qualnet. Qualnet essaye de simuler le plus fidèlement possible un réseau. Néanmoins, il y a certaines limites. Qualnet ne peut pas

reproduire le comportement du vrai matériel. En effet, il ne peut pas reproduire un environnement complexe tel qu'un hangar ou un atelier. Par exemple, dans le cas de l'atelier de l'industrie du futur, la prise en compte des interférences est importante. Nous avons envisagé deux types d'implémentation. Le premier type consiste en une extension à Quagga ou OLSRd. L'avantage est que les suites de routage fournissent un ensemble de composants logiciels facilitant le routage. Un inconvénient est qu'il nécessite un système d'exploitation complet. Le second type d'implémentation est une implémentation directement liée au noyau comme le noyau Linux. L'avantage est qu'elle peut être disponible sur plus de terminaux IoT. Le principal inconvénient est qu'elle utilise l'interface de programmation du noyau. Dans les deux cas, l'objectif aurait été d'évaluer les performances de Q-routing dans des scénarios en situation réelle. Nous avions identifié des plateformes comme Fit-IoT-Lab pour effectuer nos expérimentations.

Expérimentation sur IEEE 802.15.4

Nous avons fait l'essentiel de nos simulations avec des réseaux utilisant IEEE 802.11. IEEE 802.11 n'est pas la seule technologie candidate pour les réseaux industriels sans-fil, il y a aussi IEEE 802.15.4 (Section 2.3.1). Le passage d'IEEE 802.11 vers IEEE 802.15.4 nécessite quelques adaptations. En effet, les deux protocoles n'ont pas le même MTU (*Maximum Transmission Unit*). Le MTU d'IEEE 802.15.4 est beaucoup plus petit. Q-routing doit donc transporter moins de routes par paquet. Ceci augmente donc le nombre de paquets pour chaque mise à jour. Dans le Chapitre 5, nous avons déterminé qu'un nœud avec quatre voisins sur la grille irrégulière de Boyan et Littman [BL94] envoyait déjà jusqu'à cinq paquets par mise à jour périodique. Avec la réduction du MTU imposée par IEEE 802.15.4, ce nombre pourrait augmenter d'un facteur 10. De plus, le débit d'IEEE 802.15.4 (WPAN) est nettement inférieur à celui d'IEEE 802.11 (WLAN). La combinaison de ces deux facteurs rend notre implémentation actuelle difficilement utilisable sur une grande topologie IEEE 802.15.4.

Il existe plusieurs solutions pour résoudre ce problème. La première consiste à transformer Q-routing en protocole réactif comme AODV (RFC 3561). En effet, il n'y aurait plus de mise à jour périodique conséquente. Le trafic lié au protocole de routage serait réduit à une inondation pour chaque demande de route et un trafic périodique pour vérifier l'état de la route. Q-routing aurait aussi les mêmes qualités, mais aussi les mêmes limites que les autres protocoles de routage : il perdrait en efficacité si les changements de topologie sont trop fréquents.

La seconde solution consiste à simplement réduire le nombre de routes envoyées. Dans le cas du nœud à quatre voisins, sur les quatre routes par destination envoyées, il y aurait au mieux une seule mise à jour par destination sur le nœud receveur. Comme Q-routing met à jour les informations de la route en fonction du numéro de séquence puis de la Q-valeur de la mise à jour. Le nœud émetteur peut envoyer uniquement la route ayant le numéro de séquence le plus grand pour chaque destination. Si deux routes pour une même destination ont le même numéro de séquence, il peut envoyer la route ayant la plus faible Q-valeur. Dans le précédent exemple de notre nœud à quatre voisins sur la grille sans-fil, cela revient à envoyer deux paquets au lieu de cinq.

Production scientifique

Conférences internationales avec comité de lecture

Alexis Bitaillou et al. « Latency filtering for Q-routing on wireless networks ». In : *2021 International Wireless Communications and Mobile Computing (IWCMC)*. ISSN : 2376-6506. Juin 2021, p. 1314-1319. DOI : 10.1109/IWCMC51323.2021.9498737

Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. « New Results on Q-Routing Protocol for Wireless Networks ». In : *Ad Hoc Networks*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham : Springer International Publishing, 2021, p. 29-43. ISBN : 978-3-030-67369-7. DOI : 10.1007/978-3-030-67369-7_3

Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. « Q-routing : From the Algorithm to the Routing Protocol ». In : *Machine Learning for Networking*. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2020, p. 58-69. ISBN : 978-3-030-45778-5. DOI : 10.1007/978-3-030-45778-5_5

Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. « How to bring machine learning in industrial networks ? » In : *Fifth Sino-French Workshop on Information and Communication Technologies, SIFWICT 2019*. Nantes, France, juin 2019. URL : https://hal.archives-ouvertes.fr/hal-02161147

Conférences nationales avec comité de lecture

Alexis Bitaillou et al. « Sur un filtrage de latences pour du routage par renforcement sur des réseaux sans-fil ». In : *CORES 2021 – 6ème Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*. La Rochelle, France, sept. 2021. URL : https://hal.archives-ouvertes.fr/hal-03219086

Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. « Routage et apprentissage par renforcement ». Poster. Journées du GDR-RSD. Nantes, France, jan. 2020. URL : https://hal.archives-ouvertes.fr/hal-02536945

Autres publications

Alexis Bitaillou. « Sur un filtrage de latences pour du routage par renforcement sur des réseaux sans-fil ». In : *Journée des doctorants de l'université de Nantes*. Nantes, France, mai 2021, p. 4

Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. *Synthèse sur les protocoles de communication pour l'Internet des objets de l'industrie 4.0*. Livrable projet COWIN. LS2N, Université de Nantes ; IETR, Université de Nantes, jan. 2019. URL : `https://hal.archives-ouvertes.fr/hal-02365063`

Productions logicielles

Implémentation de Q-routing et scénarios

`https://gitlab.univ-nantes.fr/ls2n-rio/qrouting-qualnet`
Afin de rendre notre travail reproductible, l'implémentation et les scénarios présentés sont disponibles sur ce dépôt git. Le dépôt contient les différentes versions de Q-routing utilisées pour chaque publication. L'implémentation de Q-routing a été réalisée en C++ et est composée à elle seule de 1705 SLOCs dans sa dernière version.

Scripts d'automatisations et de générations des figures

`https://gitlab.univ-nantes.fr/ls2n-rio/qualnet-script`
Ce dépôt git contient les scripts nécessaires pour automatiser une partie des simulations pour en réaliser un grand nombre avec différents jeux de paramètres. Ce dépôt comprend aussi des scripts permettant de générer différentes illustrations comme les graphiques. L'ensemble a été développé en Python 3 et est composé de 1260 SLOCs.

Références bibliographiques

[Ada+14] T. Adame et al. « IEEE 802.11AH : the WiFi approach for M2M communications ». In : *IEEE Wireless Communications* 21.6 (déc. 2014), p. 144-152. ISSN : 1536-1284. DOI : 10.1109/MWC.2014.7000982 (cf. p. 28).

[Aij20] Adnan Aijaz. « Private 5G : The Future of Industrial Wireless ». In : *IEEE Industrial Electronics Magazine* 14.4 (déc. 2020). Conference Name : IEEE Industrial Electronics Magazine, p. 136-145. ISSN : 1941-0115. DOI : 10.1109/MIE.2020.3004975 (cf. p. 29, 36).

[Al +09] Khaldoun Al Agha et al. « Cross-layering in an Industrial Wireless Sensor Network : Case Study of OCARI. » In : *JNW* 4.6 (2009), p. 411-420 (cf. p. 40).

[Bak05] N. Baker. « ZigBee and Bluetooth strengths and weaknesses for industrial applications ». In : *Computing Control Engineering Journal* 16.2 (avr. 2005), p. 20-25. ISSN : 0956-3385. DOI : 10.1049/cce:20050204 (cf. p. 24).

[BCD06] Eleonora Borgia, Marco Conti et Franca Delmastro. « Mobileman : design, integration, and experimentation of cross-layer mobile multihop ad hoc networks ». In : *Communications Magazine, IEEE* 44.7 (2006), p. 80-85 (cf. p. 40).

[Bel15] Anne-Sophie Bellaiche. « En détail : Les 10 solutions de la Nouvelle France industrielle ». In : (mai 2015). ISSN : 0042-126X. URL : https://www.usinenouvelle.com/editorial/en-detail-les-10-solutions-de-la-nouvelle-france-industrielle.N329987 (visité le 07/07/2021) (cf. p. 15).

[Bel16] B. Bellalta. « IEEE 802.11ax : High-efficiency WLANS ». In : *IEEE Wireless Communications* 23.1 (fév. 2016), p. 38-46. ISSN : 1536-1284. DOI : 10.1109/MWC.2016.7422404 (cf. p. 27).

[Ber+08] M. Bertin et al. « OCARI for industrial wireless sensor networks ». In : *2008 1st IFIP Wireless Days*. 2008 1st IFIP Wireless Days. Nov. 2008, p. 1-5. DOI : 10.1109/WD.2008.4812928 (cf. p. 40, 41).

[BHC04] M. Biggs, A. Henley et T. Clarkson. « Occupancy analysis of the 2.4 GHz ISM band ». In : *IEE Proceedings - Communications* 151.5 (oct. 2004), p. 481-488. ISSN : 1350-2425. DOI : 10.1049/ip-com:20040653 (cf. p. 36).

[Bit+21a] Alexis BITAILLOU et al. « Latency filtering for Q-routing on wireless networks ». In : *2021 International Wireless Communications and Mobile Computing (IWCMC)*. ISSN : 2376-6506. Juin 2021, p. 1314-1319. DOI : 10.1109/IWCMC51323.2021.9498737 (cf. p. 127).

[Bit+21b] Alexis BITAILLOU et al. « Sur un filtrage de latences pour du routage par renforcement sur des réseaux sans-fil ». In : *CORES 2021 – 6ème Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*. La Rochelle, France, sept. 2021. URL : https://hal.archives-ouvertes.fr/hal-03219086 (cf. p. 127).

[Bit21] Alexis BITAILLOU. « Sur un filtrage de latences pour du routage par renforcement sur des réseaux sans-fil ». In : *Journée des doctorants de l'université de Nantes*. Nantes, France, mai 2021, p. 4 (cf. p. 128).

[BL94] Justin A. BOYAN et Michael L. LITTMAN. « Packet routing in dynamically changing networks : A reinforcement learning approach ». In : *Advances in neural information processing systems*. 1994, p. 671-678 (cf. p. 8, 53, 59, 60, 62, 67, 73, 75, 78, 81, 82, 88, 126).

[BM11] Venkataramana BADARLA et C. Siva Ram MURTHY. « Learning-TCP : A stochastic approach for efficient update in TCP congestion window in ad hoc wireless networks ». In : *Journal of Parallel and Distributed Computing*. Special Issue on Cloud Computing 71.6 (juin 2011), p. 863-878. ISSN : 0743-7315. DOI : 10.1016/j.jpdc.2010.12.012. URL : https://www.sciencedirect.com/science/article/pii/S0743731510002819 (cf. p. 58).

[BMS13] Milind M. BUDDHIKOT, Scott C. MILLER et Anand Prabhu SUBRAMANIAN. « Interference aware routing in multi-radio wireless mesh networks ». Brev. amér. 8532023B2. Alcatel-Lucent SAS. 10 sept. 2013. URL : https://patents.google.com/patent/US8532023B2/en (cf. p. 34).

[Bon+98] Eric BONABEAU et al. « Routing in telecommunications networks with ant-like agents ». In : *Intelligent Agents for Telecommunication Applications*. Sous la dir. de Sahin ALBAYRAK et Francisco J. GARIJO. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1998, p. 60-71. ISBN : 978-3-540-69102-0 (cf. p. 59).

[Bou+18] Raouf BOUTABA et al. « A comprehensive survey on machine learning for networking : evolution, applications and research opportunities ». In : *Journal of Internet Services and Applications* 9.1 (21 juin 2018), p. 16. ISSN : 1869-0238. DOI : 10.1186/s13174-018-0087-2. URL : https://doi.org/10.1186/s13174-018-0087-2 (cf. p. 55).

[Bou+20] S. E. BOUZID et al. « Efficient Routing Protocol for Wireless Sensor Network based on Reinforcement Learning ». In : *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. ISSN : 2687-878X. Sept. 2020, p. 1-5. DOI : 10.1109/ATSIP49331.2020.9231883 (cf. p. 66).

[BOW00] G. E. Bottomley, T. Ottosson et Y.- E. Wang. « A generalized RAKE receiver for interference suppression ». In : *IEEE Journal on Selected Areas in Communications* 18.8 (août 2000), p. 1536-1545. ISSN : 0733-8716. DOI : 10.1109/49.864017 (cf. p. 32).

[BPA19a] Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. « How to bring machine learning in industrial networks ? » In : *Fifth Sino-French Workshop on Information and Communication Technologies, SIFWICT 2019*. Nantes, France, juin 2019. URL : https://hal.archives-ouvertes.fr/hal-02161147 (cf. p. 127).

[BPA19b] Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. *Synthèse sur les protocoles de communication pour l'Internet des objets de l'industrie 4.0*. Livrable projet COWIN. LS2N, Université de Nantes ; IETR, Université de Nantes, jan. 2019. URL : https://hal.archives-ouvertes.fr/hal-02365063 (cf. p. 128).

[BPA20a] Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. « Q-routing : From the Algorithm to the Routing Protocol ». In : *Machine Learning for Networking*. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2020, p. 58-69. ISBN : 978-3-030-45778-5. DOI : 10.1007/978-3-030-45778-5_5 (cf. p. 127).

[BPA20b] Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. « Routage et apprentissage par renforcement ». Poster. Journées du GDR-RSD. Nantes, France, jan. 2020. URL : https://hal.archives-ouvertes.fr/hal-02536945 (cf. p. 127).

[BPA21] Alexis Bitaillou, Benoît Parrein et Guillaume Andrieux. « New Results on Q-Routing Protocol for Wireless Networks ». In : *Ad Hoc Networks*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham : Springer International Publishing, 2021, p. 29-43. ISBN : 978-3-030-67369-7. DOI : 10.1007/978-3-030-67369-7_3 (cf. p. 127).

[Bro01] P. Brooks. « Ethernet/IP-industrial protocol ». In : *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597)*. ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597). T. 2. Oct. 2001, 505-514 vol.2. DOI : 10.1109/ETFA.2001.997725 (cf. p. 23).

[Bru07] Joseph Bruno. « Interference Reduction in Wireless Networks ». In : *Computing Research Topics, Computing Sciences Department, Villanova University* (2007) (cf. p. 34).

[CA11] Thomas Chen et Saeed Abu-Nimeh. « Lessons from stuxnet ». In : *Computer* 44.4 (2011), p. 91-93 (cf. p. 17).

[Car+19] Thierry Carlotti et al. « An OCARI-Based Wireless Sensor Network for Heat Measurements during Outdoor Fire Experiments ». In : *Sensors* 19.1 (jan. 2019), p. 158. DOI : 10.3390/s19010158. URL : https://www.mdpi.com/1424-8220/19/1/158 (cf. p. 41, 69).

[Car08] Dick CARO. *Wireless networks for industrial automation*. OCLC : 670630160. Research Triangle Park, NC : ISA, 2008. ISBN : 978-1-61583-562-1. URL : http://app.knovel.com/hotlink/toc/id:kpWNIAE00A/wireless-networks-for (visité le 23/11/2018) (cf. p. 25, 26).

[CBV04] P. CHATZIMISIOS, A. C. BOUCOUVALAS et V. VITSAS. « Effectiveness of RTS/CTS handshake in IEEE 802.11a Wireless LANs ». In : *Electronics Letters* 40.14 (juill. 2004), p. 915-916. ISSN : 0013-5194. DOI : 10.1049/el:20040510 (cf. p. 33).

[CGT05] Marco CONTI, Enrico GREGORI et Giovanni TURI. « A Cross-layer Optimization of Gnutella for Mobile Ad Hoc Networks ». In : *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc '05. event-place : Urbana-Champaign, IL, USA. New York, NY, USA : ACM, 2005, p. 343-354. ISBN : 978-1-59593-004-0. DOI : 10.1145/1062689.1062733. URL : http://doi.acm.org/10.1145/1062689.1062733 (cf. p. 39).

[Chr17] CHRISTINE KERDELLANT. « La France doit embrasser en même temps les industries 3.0 et 4.0 - Editorial ». In : *l'Usine Nouvelle* 3509 (30 mars 2017). ISSN : 0042-126X. URL : https://www.usinenouvelle.com/editorial/la-france-doit-embrasser-en-meme-temps-les-industries-3-0-et-4-0.N519464 (visité le 22/10/2018) (cf. p. 14).

[CIS17] CISCO. *Time-Sensitive Networking : A Technical Introduction White Paper*. Rapport technique. CISCO, 2017, p. 8. URL : https://www.cisco.com/c/dam/en/us/solutions/collateral/industry-solutions/white-paper-c11-738950.pdf (visité le 09/01/2019) (cf. p. 21).

[Cla+03] David D. CLARK et al. « A knowledge plane for the internet ». In : *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, p. 3-10 (cf. p. 42).

[CNM10] Deji CHEN, Mark NIXON et Aloysius MOK. « Why WirelessHART ». In : *WirelessHART™ : Real-Time Mesh Network for Industrial Automation*. Sous la dir. de Deji CHEN, Mark NIXON et Aloysius MOK. Boston, MA : Springer US, 2010, p. 195-199. ISBN : 978-1-4419-6047-4. DOI : 10.1007/978-1-4419-6047-4_15. URL : https://doi.org/10.1007/978-1-4419-6047-4_15 (cf. p. 26).

[Con+04] M. CONTI et al. « Cross-layering in mobile ad hoc network design ». In : *Computer* 37.2 (fév. 2004), p. 48-51. ISSN : 0018-9162. DOI : 10.1109/MC.2004.1266295 (cf. p. 39).

[Con18a] CONTRIBUTEURS DE WIKIPÉDIA. *Ethernet frame*. In : *Wikipedia*. Page Version ID : 874358840. 18 déc. 2018. URL : https://en.wikipedia.org/w/index.php?title=Ethernet_frame&oldid=874358840 (visité le 19/12/2018) (cf. p. 21).

[Con18b] CONTRIBUTEURS DE WIKIPÉDIA. *SERCOS III*. In : *Wikipedia*. Page Version ID : 855929777. 21 août 2018. URL : https://en.wikipedia.org/w/index.php?title=SERCOS_III&oldid=855929777 (visité le 19/12/2018) (cf. p. 23).

[Cor97] Microsoft CORPORATION, éd. *PC 97 hardware design guide : the technical reference for designing PCs and peripherals for the Microsoft Windows family of operating systems*. Microsoft professional editions. OCLC : 833181213. Redmond, Wash : Microsoft Press, 1997. 427 p. ISBN : 978-1-57231-381-1 (cf. p. 20).

[CVV08] G. CENA, A. VALENZANO et S. VITTURI. « Hybrid wired/wireless networks for real-time communications ». In : *IEEE Industrial Electronics Magazine* 2.1 (mars 2008), p. 8-20. ISSN : 1932-4529. DOI : 10.1109/MIE.2008.917155 (cf. p. 23).

[CY96] Samuel PM CHOI et Dit-Yan YEUNG. « Predictive Q-routing : A memory-based reinforcement learning approach to adaptive traffic control ». In : *Advances in Neural Information Processing Systems*. 1996, p. 945-951 (cf. p. 63, 67, 70, 78, 91, 122).

[CYC16] Hao CHEN, Yue YIN et Ge CHEN. *Cross-layer Cooperation for Encrypted Traffic*. Internet-Draft draft-chen-tsvwg-crosslayer-cooperation-00. Work in Progress, Expired & archived. Internet Engineering Task Force, mars 2016. 7 p. URL : https://datatracker.ietf.org/doc/html/draft-chen-tsvwg-crosslayer-cooperation-00 (cf. p. 42).

[CYZ17] Yujun CHENG, Dong YANG et Huachun ZHOU. « Det-WiFi : A Multihop TDMA MAC Implementation for Industrial Deterministic Applications Based on Commodity 802.11 Hardware ». In : *Wireless Communications and Mobile Computing* 2017 (2017) (cf. p. 27).

[Dan+14] M. DANIELETTO et al. « CARMEN : a cognitive networking testbed on android OS devices ». In : *IEEE Communications Magazine* 52.9 (sept. 2014), p. 98-107. ISSN : 0163-6804. DOI : 10.1109/MCOM.2014.6894459 (cf. p. 49-51).

[DAS14] Domenico DE GUGLIELMO, Giuseppe ANASTASI et Alessio SEGHETTI. « From ieee 802.15.4 to ieee 802.15.4e : A step towards the internet of things ». In : *Advances onto the Internet of Things*. Springer, 2014, p. 135-152 (cf. p. 25).

[Dec09] J. DECOTIGNIE. « The Many Faces of Industrial Ethernet [Past and Present] ». In : *IEEE Industrial Electronics Magazine* 3.1 (mars 2009), p. 8-19. ISSN : 1932-4529. DOI : 10.1109/MIE.2009.932171 (cf. p. 20, 69).

[DP81] Yogen K. DALAL et Robert S. PRINTIS. « 48-bit absolute internet and Ethernet host numbers ». In : *ACM SIGCOMM Computer Communication Review*. T. 11. ACM, 1981, p. 240-245 (cf. p. 21).

[DW93] Wim DIEPSTRATEN et N. WCND-UTRECHT. « IEEE 802.11 wireless access method and physical specification ». In : *Power* 5.10 (1993) (cf. p. 27).

[EE04] T. ELBATT et A. EPHREMIDES. « Joint scheduling and power control for wireless ad hoc networks ». In : *IEEE Transactions on Wireless Communications* 3.1 (jan. 2004), p. 74-85. ISSN : 1536-1276. DOI : 10.1109/TWC.2003.819032 (cf. p. 37).

[Ehr17] Jesse M. EHRENFELD. « WannaCry, Cybersecurity and Health Information Technology : A Time to Act ». In : *Journal of Medical Systems* 41.7 (24 mai 2017), p. 104. ISSN : 1573-689X. DOI : 10.1007/s10916-017-0752-1. URL : https://doi.org/10.1007/s10916-017-0752-1 (cf. p. 17).

[Fal+19] Filipe FALCÃO et al. « Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection ». In : *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC '19. New York, NY, USA : Association for Computing Machinery, avr. 2019, p. 318-327. ISBN : 978-1-4503-5933-7. DOI : 10.1145/3297280.3297314. URL : https://doi.org/10.1145/3297280.3297314 (cf. p. 53).

[Fel05] M. FELSER. « Real-Time Ethernet - Industry Prospective ». In : *Proceedings of the IEEE* 93.6 (juin 2005), p. 1118-1129. ISSN : 0018-9219. DOI : 10.1109/JPROC.2005.849720 (cf. p. 20, 22).

[Fer02] Niels FERGUSON. « Michael : an improved MIC for 802.11 WEP ». In : *IEEE doc* 802.2 (2002) (cf. p. 28).

[Flo+13] A. B. FLORES et al. « IEEE 802.11af : a standard for TV white space spectrum sharing ». In : *IEEE Communications Magazine* 51.10 (oct. 2013), p. 92-100. ISSN : 0163-6804. DOI : 10.1109/MCOM.2013.6619571 (cf. p. 28).

[FMS01] Scott FLUHRER, Itsik MANTIN et Adi SHAMIR. « Weaknesses in the Key Scheduling Algorithm of RC4 ». In : *Selected Areas in Cryptography*. Sous la dir. de Serge VAUDENAY et Amr M. YOUSSEF. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, p. 1-24. ISBN : 978-3-540-45537-0 (cf. p. 28).

[För+08] A. FÖRSTER et al. « An Efficient Implementation of Reinforcement Learning Based Routing on Real WSN Hardware ». In : *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. Oct. 2008, p. 247-252. DOI : 10.1109/WiMob.2008.99 (cf. p. 59).

[FS02] M. FELSER et T. SAUTER. « The fieldbus war : history or short break between battles ? » In : *4th IEEE International Workshop on Factory Communication Systems*. 4th IEEE International Workshop on Factory Communication Systems. Août 2002, p. 73-80. DOI : 10.1109/WFCS.2002.1159702 (cf. p. 20).

[GKM96] Patrick GOETZ, Shailesh KUMAR et Risto MIIKKULAINEN. « On-Line Adaptation of a Signal Predistorter through Dual Reinforcement Learning ». In : *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*. ICML'96. Bari, Italy : Morgan Kaufmann Publishers Inc., 1996, p. 175-181. ISBN : 1558604197 (cf. p. 63).

[Gon05] Michelle Xiaohong GONG. « Improving the Capacity in Wireless Ad Hoc Networks through Multiple Channel Operation : Design Principles and Protocols ». Thèse de doct. Blacksburg, USA : Faculty of the Virginia Polytechnic Institute et State University, 7 juin 2005. 162 p. URL : https://vtechworks.lib.vt.edu/handle/10919/28052 (cf. p. 37).

[Gri17] D. A. Grier. « The Radical Technology of Industrie 4.0 ». In : *Computer* 50.4 (avr. 2017), p. 120-120. ISSN : 0018-9162. DOI : 10.1109/MC.2017.109 (cf. p. 16, 121).

[Has10] Hado van Hasselt. « Double Q-learning ». In : *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*. NIPS'10. event-place : Vancouver, British Columbia, Canada. USA : Curran Associates Inc., 2010, p. 2613-2621. URL : http://dl.acm.org/citation.cfm?id=2997046.2997187 (cf. p. 56, 57, 125).

[HCL18] Thomas Hendriks, Miguel Camelo et Steven Latré. « Q2-Routing : A Qos-aware Q-Routing algorithm for Wireless Ad Hoc Networks ». In : *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. ISSN : 2160-4886. Oct. 2018, p. 108-115. DOI : 10.1109/WiMOB.2018.8589161 (cf. p. 64, 67, 75).

[HF10] T. Hu et Y. Fei. « QELAR : A Machine-Learning-Based Adaptive Routing Protocol for Energy-Efficient and Lifetime-Extended Underwater Sensor Networks ». In : *IEEE Transactions on Mobile Computing* 9.6 (juin 2010), p. 796-809. ISSN : 1536-1233. DOI : 10.1109/TMC.2010.28 (cf. p. 60).

[HGS16] Hado van Hasselt, Arthur Guez et David Silver. « Deep Reinforcement Learning with Double Q-Learning ». In : *Thirtieth AAAI Conference on Artificial Intelligence*. Thirtieth AAAI Conference on Artificial Intelligence. 2 mars 2016. URL : https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12389 (cf. p. 58).

[Hoc04] Said Hoceini. « Techniques d'Apprentissage par Renforcement pour le Routage Adaptatif dans les Réseaux de Télécommunication à Trafic Irrégulie ». Thèse de doct. Université Paris XII Val de Marne, nov. 2004. URL : https://tel.archives-ouvertes.fr/tel-00010430 (cf. p. 64, 67, 75).

[IEE+12] IEEE Computer Society et al. *IEEE standard for local and metropolitan area networks. Part 15.4, Amendment 3, Part 15.4, Amendment 3,* OCLC : 800045409. New York : Institute of Electrical et Electronics Engineers, 2012. ISBN : 978-0-7381-7259-0. URL : http://ieeexplore.ieee.org/servlet/opac?punumber=6190696 (cf. p. 24).

[IEE+13] IEEE Computer Society et al. *IEEE standard for local and metropolitan area networks. Amendment 5 : Physical layer specifications for low energy, critical infrastructure monitoring networks Part 15.4, Part 15.4,* OCLC : 858036474. 2013. ISBN : 978-0-7381-8446-3. URL : http://ieeexplore.ieee.org/servlet/opac?punumber=6581826 (cf. p. 24).

[IEE+99] IEEE Computer Society et al. *Supplement to IEEE standard for Information technology–telecommunications and information exchange between systems– local and metropolitan area networks – specific requirements : part 11 : wireless LAN medium access control (MAC) and physical layer (PHY) specifications : High-speed physical layer in the 5 GHz band*. OCLC : 50293188. New York, N.Y., USA : Institute of Electrical et Electronics

Engineers, 1999. ISBN : 978-0-7381-1809-3. URL : http://ieeexplore.ieee.org/lpdocs/epic03/standards.htm (cf. p. 27).

[IEE03] IEEE Computer Society. *IEEE Std 802.15.4-2003 : IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (.* OCLC : 956670333. S.l. : IEEE, 2003. ISBN : 978-0-7381-3686-8. URL : http://ieeexplore.ieee.org/servlet/opac?punumber=8762 (cf. p. 24).

[IEE05] IEEE Computer Society. « IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8 : Medium Access Control (MAC) Quality of Service Enhancements ». In : *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)* (nov. 2005), p. 1-212. DOI : 10.1109/IEEESTD.2005.97890 (cf. p. 27).

[IL11] IEEE Computer Society et LAN/MAN Standards Committee. « IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks ». In : *IEEE Std 802.1AS-2011* (mars 2011), p. 1-292. DOI : 10.1109/IEEESTD.2011.5741898 (cf. p. 21).

[IL97] IEEE Computer Society et LAN/MAN Standards Committee. *Information technology – telecommunications and information exchange between systems – local and metropolitan area networks – specific requirements – part 11, wireless LAN medium access control (MAC) and physical layer (PHY) specifications*. OCLC : 38598622. New York, N.Y. : Institute of Electrical et Electronics Engineers, 1997. ISBN : 978-1-55937-935-9 (cf. p. 27).

[JA07] H. Jasani et N. Alaraje. « Evaluating the performance of IEEE 802.11 network using RTS/CTS mechanism ». In : *2007 IEEE International Conference on Electro/Information Technology*. 2007 IEEE International Conference on Electro/Information Technology. Mai 2007, p. 616-621. DOI : 10.1109/EIT.2007.4374535 (cf. p. 33).

[JB04] D. Jansen et H. Buttner. « Real-time ethernet the EtherCAT solution ». In : *Computing Control Engineering Journal* 15.1 (fév. 2004), p. 16-21. ISSN : 0956-3385. DOI : 10.1049/cce:20040104 (cf. p. 22).

[Jia+17] Hong Jiang et al. « TCP-Gvegas with prediction and adaptation in multi-hop ad hoc networks ». In : *Wireless Networks* 23.5 (1er juill. 2017), p. 1535-1548. ISSN : 1572-8196. DOI : 10.1007/s11276-016-1242-y. URL : https://doi.org/10.1007/s11276-016-1242-y (cf. p. 59).

[Kag13] Henning Kagermann. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 : Securing the Future of German Manufacturing Industry ; Final Report of the Industrie 4.0 Working Group*. Forschungsunion, avr. 2013, p. 82 (cf. p. 11, 15, 16, 69, 121).

[Kar90] Phil KARN. « MACA-a new channel access method for packet radio ». In : *ARRL/CRRL Amateur radio 9th computer networking conference*. T. 140. London, Canada, 1990, p. 134-140 (cf. p. 33).

[KAT06] A. KOUBAA, M. ALVES et E. TOVAR. « GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks ». In : *Proceedings 20th IEEE International Parallel Distributed Processing Symposium*. Proceedings 20th IEEE International Parallel Distributed Processing Symposium. Avr. 2006, 8 p. DOI : 10.1109/IPDPS.2006.1639415 (cf. p. 25).

[KDI10] M. KNEZIC, B. DOKIC et Z. IVANOVIC. « Topology aspects in EtherCAT networks ». In : *Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010*. Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010. Sept. 2010, T1–1–T1-6. DOI : 10.1109/EPEPEMC.2010.5606688 (cf. p. 21, 22).

[KE97] A. KAMERMAN et N. ERKOCEVIC. « Microwave oven interference on wireless LANs operating in the 2.4 GHz ISM band ». In : *Proceedings of 8th International Symposium on Personal, Indoor and Mobile Radio Communications - PIMRC '97*. Proceedings of 8th International Symposium on Personal, Indoor and Mobile Radio Communications - PIMRC '97. T. 3. Sept. 1997, 1221-1227 vol.3. DOI : 10.1109/PIMRC.1997.627080 (cf. p. 36).

[Kho+15] Evgeny KHOROV et al. « A survey on IEEE 802.11ah : An enabling networking technology for smart cities ». In : *Computer Communications*. Special Issue on Networking and Communications for Smart Cities 58 (1er mars 2015), p. 53-69. ISSN : 0140-3664. DOI : 10.1016/j.comcom.2014.08.008. URL : http://www.sciencedirect.com/science/article/pii/S0140366414002989 (cf. p. 28).

[Kim+08] A. N. KIM et al. « When HART goes wireless : Understanding and implementing the WirelessHART standard ». In : *2008 IEEE International Conference on Emerging Technologies and Factory Automation*. 2008 IEEE International Conference on Emerging Technologies and Factory Automation. Sept. 2008, p. 899-907. DOI : 10.1109/ETFA.2008.4638503 (cf. p. 26).

[KK05] V. KAWADIA et P. R. KUMAR. « A cautionary perspective on cross-layer design ». In : *IEEE Wireless Communications* 12.1 (fév. 2005), p. 3-11. ISSN : 1536-1284. DOI : 10.1109/MWC.2005.1404568 (cf. p. 42).

[Kom+10] R. S. KOMALI et al. « The price of ignorance : distributed topology control in cognitive networks ». In : *IEEE Transactions on Wireless Communications* 9.4 (avr. 2010), p. 1434-1445. ISSN : 1536-1276. DOI : 10.1109/TWC.2010.04.090400 (cf. p. 44).

[Kum97] Shailesh KUMAR. « Dual Reinforcement Q-routing : An On-line Adaptive Routing Algorithm ». In : *Proceedings of the Articial Neural Networks in Engineering Conference, 1997* (1997) (cf. p. 63, 78).

[Kus13] D. KUSHNER. « The real story of stuxnet ». In : *IEEE Spectrum* 50.3 (mars 2013), p. 48-53. ISSN : 0018-9235. DOI : 10.1109/MSPEC.2013.6471059 (cf. p. 17).

[Lau+17] M. LAURIDSEN et al. « Interference Measurements in the European 868 MHz ISM Band with Focus on LoRa and SigFox ». In : *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. 2017 IEEE Wireless Communications and Networking Conference (WCNC). Mars 2017, p. 1-6. DOI : 10.1109/WCNC.2017.7925650 (cf. p. 35).

[LBV08] Erwan LIVOLANT, Adrien van den BOSSCHE et Thierry VAL. « MAC specifications for a WPAN allowing both energy saving and guaranteed delay* ». In : *Wireless Sensor and Actor Networks II*. Sous la dir. d'Ali MIRI. IFIP – The International Federation for Information Processing. Springer US, 2008, p. 233-244. ISBN : 978-0-387-09441-0 (cf. p. 40).

[LC04] Xuefei LI et L. CUTHBERT. « On-demand node-disjoint multipath routing in wireless ad hoc networks ». In : *29th Annual IEEE International Conference on Local Computer Networks*. 29th Annual IEEE International Conference on Local Computer Networks. Nov. 2004, p. 419-420. DOI : 10.1109/LCN.2004.90 (cf. p. 34).

[LCS12] Y. LIU, L. X. CAI et X. S. SHEN. « Spectrum-Aware Opportunistic Routing in Multi-Hop Cognitive Radio Networks ». In : *IEEE Journal on Selected Areas in Communications* 30.10 (nov. 2012), p. 1958-1968. ISSN : 0733-8716. DOI : 10.1109/JSAC.2012.121111 (cf. p. 39).

[Lee+07a] Minsoo LEE et al. *Autonomic Reconfiguration Management for Heterogeneous Wireless Networks using Reinforcement Learning*. Rapport technique. Davis, CA 95616, USA : Department of Electrical et Computer Engineering, University of California Davis, 2007, p. 8. URL : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.509.915&rep=rep1&type=pdf (cf. p. 46, 50).

[Lee+07b] Minsoo LEE et al. « Cognitive Network Management with Reinforcement Learning for Wireless Mesh Networks ». In : *IP Operations and Management*. Sous la dir. de Deep MEDHI et al. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, p. 168-179. ISBN : 978-3-540-75853-2 (cf. p. 43, 44).

[Lee+11] Young LEE et al. *Problem Statement for Cross-Layer Optimization*. Internet-Draft draft-lee-cross-layer-optimization-problem-02. Work in Progress, Expired & archived. Internet Engineering Task Force, jan. 2011. 22 p. URL : https://datatracker.ietf.org/doc/html/draft-lee-cross-layer-optimization-problem-02 (cf. p. 42).

[Leg15] LEGIFRANCE. *Décision n° 2014-1263 du 6 novembre 2014 fixant les conditions d'utilisation des fréquences radioélectriques par des dispositifs à courte portée*. 30 jan. 2015. URL : https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000030160927 (cf. p. 34).

[Li+15] Qiang LI et al. « Kalman Filter and Its Application ». In : *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. Nov. 2015, p. 74-77. DOI : 10.1109/ICINIS.2015.35 (cf. p. 109).

[Li+16] Wei Li et al. « Learning-Based and Data-Driven TCP Design for Memory-Constrained IoT ». In : *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*. ISSN : 2325-2944. Mai 2016, p. 199-205. DOI : 10.1109/DCOSS.2016.8 (cf. p. 59).

[Lia+11] Wei Liang et al. « Survey and experiments of WIA-PA specification of industrial wireless network ». In : *Wireless Communications and Mobile Computing* 11.8 (1er août 2011), p. 1197-1212. ISSN : 1530-8677. DOI : 10.1002/wcm.976. URL : https://onlinelibrary.wiley.com/doi/abs/10.1002/wcm.976 (cf. p. 25).

[Lia+13] Wei Liang et al. « Research of Adaptive Frequency Hopping Technology in WIA-PA Industrial Wireless Network ». In : *Advances in Wireless Sensor Networks*. Sous la dir. de Ruchuan Wang et Fu Xiao. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2013, p. 248-262. ISBN : 978-3-642-36252-1 (cf. p. 25).

[LM12] Demian Lekomtcev et Roman Maršálek. « Comparison of 802.11 af and 802.22 standards–physical layer and cognitive functionality ». In : *Elektro Revue* 3.2 (2012), p. 12-18 (cf. p. 28).

[LPN11] P. H. Le, G. Pujolle et T. Nguyen. « An interference-aware multi-path routing protocol for mobile ad hoc network ». In : *2011 International Conference on Networking, Sensing and Control*. 2011 International Conference on Networking, Sensing and Control. Avr. 2011, p. 503-507. DOI : 10.1109/ICNSC.2011.5874934 (cf. p. 34).

[Luo+10] C. Luo et al. « Cross-Layer Design for TCP Performance Improvement in Cognitive Radio Networks ». In : *IEEE Transactions on Vehicular Technology* 59.5 (juin 2010), p. 2485-2495. ISSN : 0018-9545. DOI : 10.1109/TVT.2010.2041802 (cf. p. 39).

[Mah+04] P. Mahonen et al. « Hop-by-hop toward future mobile broadband IP ». In : *IEEE Communications Magazine* 42.3 (mars 2004), p. 138-146. ISSN : 0163-6804. DOI : 10.1109/MCOM.2004.1273785 (cf. p. 43).

[Max06] Maxim Integrated Products, Inc.. *How Far and How Fast Can You Go with RS-485 ? - Application Note - Maxim*. Maxim Integrated - Analog, linear, & mixed-signal devices. 25 juill. 2006. URL : https://www.maximintegrated.com/en/app-notes/index.mvp/id/3884 (visité le 23/10/2018) (cf. p. 21).

[Mit00] Joseph Mitola. « Cognitive radio ». Thèse de doct. Institutionen för teleinformatik, 2000. 313 p. (cf. p. 43).

[MM09] Saoucene Mahfoudh et Pascale Minet. « Maximization of Energy Efficiency in Wireless ad hoc and Sensor Networks With SERENA ». In : *Mobile Information Systems* 5.1 (2009). URL : http://dx.doi.org/10.3233/MIS-2009-0071 (cf. p. 41).

[MM11] Saoucene MAHFOUDH et Pascale MINET. « EOLSR : AN ENERGY EFFICIENT ROUTING PROTOCOL IN WIRELESS AD HOC AND SENSOR NETWORKS ». In : *Journal of Interconnection Networks* (21 nov. 2011). DOI : 10.1142/S0219265908002345. URL : https://www.worldscientific.com/doi/pdf/10.1142/S0219265908002345 (cf. p. 41).

[MM99] J. MITOLA et G. Q. MAGUIRE. « Cognitive radio : making software radios more personal ». In : *IEEE Personal Communications* 6.4 (août 1999), p. 13-18. ISSN : 1070-9916. DOI : 10.1109/98.788210 (cf. p. 43).

[MN09] M. Z. MALNAR et N. J. NESKOVIĆ. « Comparison of ETX and HOP count metrics using Glomosim simulator ». In : *2009 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services*. 2009 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services. Oct. 2009, p. 85-88. DOI : 10.1109/TELSKS.2009.5339402 (cf. p. 34).

[Mni+15] Volodymyr MNIH et al. « Human-level control through deep reinforcement learning ». In : *Nature* 518.7540 (fév. 2015), p. 529-533. ISSN : 1476-4687. DOI : 10.1038/nature14236. URL : https://www.nature.com/articles/nature14236/ (cf. p. 58).

[Mol85] Michael K. MOLLOY. « Collision resolution on the CSMA/CD bus ». In : *Computer Networks and ISDN Systems*. Local Area Networks 9.3 (1er mars 1985), p. 209-214. ISSN : 0169-7552. DOI : 10.1016/0169-7552(85)90044-3. URL : http://www.sciencedirect.com/science/article/pii/0169755285900443 (cf. p. 21).

[Mon+07] G. MONTENEGRO et al. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. RFC 4944. Internet Requests for Comments. RFC Editor, 2007. URL : https://www.rfc-editor.org/info/rfc4944 (cf. p. 25).

[MRZ07a] B. S. MANOJ, R. R. RAO et M. ZORZI. « On the Use of Higher Layer Information for Cognitive Networking ». In : *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*. IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference. Nov. 2007, p. 3568-3573. DOI : 10.1109/GLOCOM.2007.678 (cf. p. 45-48, 50).

[MRZ07b] B. S. MANOJ, Ramesh R. RAO et Michele ZORZI. « Architectures and Protocols for Next Generation Cognitive Networking ». In : *Cognitive Wireless Networks : Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications*. Sous la dir. de Frank H. P. FITZEK et Marcos D. KATZ. Dordrecht : Springer Netherlands, 2007, p. 271-284. ISBN : 978-1-4020-5979-7. DOI : 10.1007/978-1-4020-5979-7_14. URL : https://doi.org/10.1007/978-1-4020-5979-7_14 (cf. p. 49, 50).

[MRZ08] B. S. MANOJ, R. R. RAO et M. ZORZI. « CogNet : a cognitive complete knowledge network system ». In : *IEEE Wireless Communications* 15.6 (déc. 2008), p. 81-88. ISSN : 1536-1284. DOI : 10.1109/MWC.2008.4749751 (cf. p. 47, 48).

[Nob03] Stéphane NOBILET. « Etude et optimisation des techniques MC-CDMA pour les futures générations de systèmes de communications hertziennes ». Thèse de doct. INSA de Rennes, 3 oct. 2003. URL : https://tel.archives-ouvertes.fr/tel-00004081/document (cf. p. 33).

[NR12] Mark NIXON et T. X. ROUND ROCK. « A Comparison of WirelessHART and ISA100. 11a ». In : *Whitepaper, Emerson Process Management* (2012), p. 1-36 (cf. p. 26).

[Orm03] H. ORMAN. « The Morris worm : a fifteen-year perspective ». In : *IEEE Security Privacy* 99.5 (sept. 2003), p. 35-43. ISSN : 1540-7993. DOI : 10.1109/MSECP.2003.1236233 (cf. p. 17).

[Pac+19] Fannia PACHECO et al. « Towards the Deployment of Machine Learning Solutions in Network Traffic Classification : A Systematic Survey ». In : *IEEE Communications Surveys Tutorials* 21.2 (2019). Conference Name : IEEE Communications Surveys Tutorials, p. 1988-2014. ISSN : 1553-877X. DOI : 10.1109/COMST.2018.2883147 (cf. p. 53).

[Pap01] Christos H. PAPADIMITRIOU. « Algorithms, Games, and the Internet ». In : *Automata, Languages and Programming*. Sous la dir. de Fernando OREJAS, Paul G. SPIRAKIS et Jan van LEEUWEN. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, p. 1-3. ISBN : 978-3-540-48224-6 (cf. p. 44).

[PC11] Stig PETERSEN et Simon CARLSEN. « WirelessHART Versus ISA100.11a : The Format War Hits the Factory Floor ». In : *IEEE Industrial Electronics Magazine* 5.4 (déc. 2011), p. 23-34. ISSN : 1932-4529. DOI : 10.1109/MIE.2011.943023. URL : http://ieeexplore.ieee.org/document/6102417/ (cf. p. 25, 26).

[PDC06] Andrea PASSARELLA, Franca DELMASTRO et Marco CONTI. « XScribe : a stateless, cross-layer approach to P2P multicast in multi-hop ad hoc networks ». In : *Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking*. ACM, 2006, p. 6-11 (cf. p. 40).

[Pet+07] M. PETROVA et al. « Interference Measurements on Performance Degradation between Colocated IEEE 802.11g/n and IEEE 802.15.4 Networks ». In : *Sixth International Conference on Networking (ICN'07)*. Sixth International Conference on Networking (ICN'07). Avr. 2007, p. 93-93. DOI : 10.1109/ICN.2007.53 (cf. p. 36, 69).

[Pry08] G. PRYTZ. « A performance analysis of EtherCAT and PROFINET IRT ». In : *2008 IEEE International Conference on Emerging Technologies and Factory Automation*. 2008 IEEE International Conference on Emerging Technologies and Factory Automation. Sept. 2008, p. 408-415. DOI : 10.1109/ETFA.2008.4638425 (cf. p. 22).

[Ram04] Chris RAMMING. « Cognitive networks ». In : *DARPATech 2004*. DARPATech 2004. Anaheim, USA, mars 2004. URL : https://webarchive.library.unt.edu/eot2008/20080923072826/http://www.arpa.gov/DARPATech2004/proceedings.html (cf. p. 43).

[RI04] Vijay T RAISINGHANI et Sridhar IYER. « Cross-layer design optimizations in wireless protocol stacks ». In : *Computer Communications*. Advances in Future Mobile/Wireless Networks and Services 27.8 (1er mai 2004), p. 720-724. ISSN : 0140-3664. DOI : 10.1016/j.comcom.2003.10.011. URL : http://www.sciencedirect.com/science/article/pii/S0140366403002913 (cf. p. 38).

[Ric+05] P. von RICKENBACH et al. « A robust interference model for wireless ad-hoc networks ». In : *19th IEEE International Parallel and Distributed Processing Symposium*. 19th IEEE International Parallel and Distributed Processing Symposium. Avr. 2005, 8 pp.–. DOI : 10.1109/IPDPS.2005.65 (cf. p. 34).

[RN94] Gavin A. RUMMERY et Mahesan NIRANJAN. *On-line Q-learning using connectionist systems*. Rapport technique 166. England : University of Cambridge, Department of Engineering Cambridge, England, sept. 1994, p. 21. URL : http://mi.eng.cam.ac.uk/reports/svr-ftp/auto-pdf/rummery_tr166.pdf (cf. p. 56).

[Rou09] Tim ROUGHGARDEN. « Intrinsic robustness of the price of anarchy ». In : *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, p. 513-522 (cf. p. 44).

[Sar07] Pasi SAROLAHTI. *Transport-layer Considerations for Explicit Cross-layer Indications*. Internet-Draft draft-sarolahti-tsvwg-crosslayer-01. Work in Progress, Expired & archived. Internet Engineering Task Force, mars 2007. 29 p. URL : https://datatracker.ietf.org/doc/html/draft-sarolahti-tsvwg-crosslayer-01 (cf. p. 42).

[SB18] Richard S. SUTTON et Andrew BARTO. *Reinforcement learning : an introduction*. Second edition. Adaptive computation and machine learning. OCLC : 1050693967. Cambridge, MA London : The MIT Press, 2018. ISBN : 978-0-262-03924-6 (cf. p. 54, 56, 57, 90).

[SBM06] Anand Prabhu SUBRAMANIAN, Milind M. BUDDHIKOT et Scott MILLER. « Interference aware routing in multi-radio wireless mesh networks ». In : *IEEE workshop on wireless mesh networks (WiMesh)*. T. 9. 2006, p. 55-63 (cf. p. 34).

[Sch01] V. SCHIFFER. « The CIP family of fieldbus protocols and its newest member - Ethernet/IP ». In : *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597)*. ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.01TH8597). Oct. 2001, 377-384 vol.1. DOI : 10.1109/ETFA.2001.996391 (cf. p. 23).

[Sch04] E. SCHEMM. « SERCOS to link with ethernet for its third generation ». In : *Computing Control Engineering Journal* 15.2 (avr. 2004), p. 30-33. ISSN : 0956-3385. DOI : 10.1049/cce:20040205 (cf. p. 23).

[Sch11] Ronald W. SCHAFER. « What Is a Savitzky-Golay Filter ? [Lecture Notes] ». In : *IEEE Signal Processing Magazine* 28.4 (juill. 2011). Conference Name : IEEE Signal Processing Magazine, p. 111-117. ISSN : 1558-0792. DOI : 10.1109/MSP.2011.941097 (cf. p. 109).

[Sem20] SEMTECH. *LoRa® and LoRaWAN® : A Technical Overview*. Rapport technique. Semtech Corporation, fév. 2020, p. 26. URL : https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf (visité le 25/05/2021) (cf. p. 33).

[Set+05] E. SETTON et al. « Cross-layer design of ad hoc networks for real-time video streaming ». In : *IEEE Wireless Communications* 12.4 (août 2005), p. 59-65. ISSN : 1536-1284. DOI : 10.1109/MWC.2005.1497859 (cf. p. 40).

[SG05] A. SIKORA et V. F. GROZA. « Coexistence of IEEE802.15.4 with other Systems in the 2.4 GHz-ISM-Band ». In : *2005 IEEE Instrumentationand Measurement Technology Conference Proceedings*. 2005 IEEE Instrumentationand Measurement Technology Conference Proceedings. T. 3. Mai 2005, p. 1786-1791. DOI : 10.1109/IMTC.2005.1604479 (cf. p. 36).

[SH80] John F. SHOCH et Jon A. HUPP. « Measured performance of an Ethernet local network ». In : *Communications of the ACM* 23.12 (1980), p. 711-721 (cf. p. 21).

[Sho+85] J. F. SHOCH et al. « The Ethernet ». In : *Local Area Networks : An Advanced Course*. Sous la dir. de D. HUTCHISON, J. A. MARIANI et W. D. SHEPHERD. Berlin, Heidelberg : Springer Berlin Heidelberg, 1985, p. 1-35. ISBN : 978-3-540-39286-6 (cf. p. 21).

[Sil+17] David SILVER et al. « Mastering the game of go without human knowledge ». In : *Nature* 550.7676 (2017), p. 354 (cf. p. 58).

[SM05] V. SRIVASTAVA et M. MOTANI. « Cross-layer design : a survey and the road ahead ». In : *IEEE Communications Magazine* 43.12 (déc. 2005), p. 112-119. ISSN : 0163-6804. DOI : 10.1109/MCOM.2005.1561928 (cf. p. 37-39).

[SNJ20] Abdellatif SERHANI, Najib NAJA et Abdellah JAMALI. « AQ-Routing : mobility-, stability-aware adaptive routing protocol for data routing in MANET–IoT systems ». In : *Cluster Computing* 23.1 (mars 2020), p. 13-27. ISSN : 1573-7543. DOI : 10.1007/s10586-019-02937-x. URL : https://doi.org/10.1007/s10586-019-02937-x (cf. p. 65, 78).

[ST16] Nicholas S. SAMARAS et Foteini S. TRIANTARI. « On Direct Diffusion Routing for Wireless Sensor Networks ». In : *2016 Advances in Wireless and Optical Communications (RTUWO)*. Nov. 2016, p. 89-94. DOI : 10.1109/RTUWO.2016.7821862 (cf. p. 59).

[SV87] A. A. M. SALEH et R. VALENZUELA. « A Statistical Model for Indoor Multipath Propagation ». In : *IEEE Journal on Selected Areas in Communications* 5.2 (fév. 1987), p. 128-137. ISSN : 0733-8716. DOI : 10.1109/JSAC.1987.1146527 (cf. p. 32).

[SVS20] Richa SHARMA, Vasudha VASHISHT et Umang SINGH. « Modelling and simulation frameworks for wireless sensor networks : a comparative study ». en. In : *IET Wireless Sensor Systems* 10.5 (mai 2020). Publisher : IET Digital Library, p. 181-197. ISSN : 2043-6394. DOI : 10.1049/iet-wss.2020.0046. URL : https://digital-library.theiet.org/content/journals/10.1049/iet-wss.2020.0046 (visité le 16/04/2021) (cf. p. 76).

[TB09] Erik TEWS et Martin BECK. « Practical attacks against WEP and WPA ». In : *Proceedings of the second ACM conference on Wireless network security*. ACM, 2009, p. 79-86 (cf. p. 28).

[TB80] Fouad A TOBAGI et V BRUCE HUNT. « Performance analysis of carrier sense multiple access with collision detection ». In : *Computer Networks (1976)*. Selected Papers from the First Symposium on Local area Communications Networks 4.5 (1er oct. 1980), p. 245-259. ISSN : 0376-5075. DOI : 10.1016/0376-5075(80)90014-8. URL : http://www.sciencedirect.com/science/article/pii/0376507580900148 (cf. p. 33).

[TC94] Ywh-Ren TSAI et Jin-Fu CHANG. « The feasibility of combating multipath interference by chirp spread spectrum techniques over Rayleigh and Rician fading channels ». In : *Proceedings of IEEE 3rd International Symposium on Spread Spectrum Techniques and Applications (ISSSTA'94)*. Proceedings of IEEE 3rd International Symposium on Spread Spectrum Techniques and Applications (ISSSTA'94). Juill. 1994, 282-286 vol.1. DOI : 10.1109/ISSSTA.1994.379577 (cf. p. 33).

[TDM05] R. W. THOMAS, L. A. DASILVA et A. B. MACKENZIE. « Cognitive networks ». In : *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.* First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. Nov. 2005, p. 352-360. DOI : 10.1109/DYSPAN.2005.1542652 (cf. p. 43, 44).

[Tho+06] R. W. THOMAS et al. « Cognitive networks : adaptation and learning to achieve end-to-end performance objectives ». In : *IEEE Communications Magazine* 44.12 (déc. 2006), p. 51-57. ISSN : 0163-6804. DOI : 10.1109/MCOM.2006.273099 (cf. p. 44, 45).

[THT08] J. TEO, Y. HA et C. THAM. « Interference-Minimized Multipath Routing with Congestion Control in Wireless Sensor Network for High-Rate Streaming ». In : *IEEE Transactions on Mobile Computing* 7.9 (sept. 2008), p. 1124-1137. ISSN : 1536-1233. DOI : 10.1109/TMC.2008.24 (cf. p. 34).

[Ute17] Martin UTERSINGER. « Le virus Petya a coûté plus d'un milliard d'euros aux entreprises ». In : *Le Monde* (7 nov. 2017). ISSN : 0395-2037. URL : https://www.lemonde.fr/pixels/article/2017/11/07/le-virus-petya-a-coute-plus-d-un-milliard-d-euros-aux-entreprises_5211421_4408996.html (visité le 27/11/2018) (cf. p. 17).

[Vej+17] B. Vejlgaard et al. « Interference Impact on Coverage and Capacity for Low Power Wide Area IoT Networks ». In : *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. 2017 IEEE Wireless Communications and Networking Conference (WCNC). Mars 2017, p. 1-6. doi : 10.1109/WCNC.2017.7925510 (cf. p. 35).

[VMS05] B. Venkata Ramana, B.S. Manoj et C. Siva Ram Murthy. « Learning-TCP : a novel learning automata based reliable transport protocol for ad hoc wireless networks ». In : *2nd International Conference on Broadband Networks, 2005.* Oct. 2005, 484-493 Vol. 1. doi : 10.1109/ICBN.2005.1589652 (cf. p. 58).

[VP17] Mathy Vanhoef et Frank Piessens. « Key reinstallation attacks : Forcing nonce reuse in WPA2 ». In : *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, p. 1313-1328 (cf. p. 28).

[VR19] Mathy Vanhoef et Eyal Ronen. *Dragonblood : A Security Analysis of WPA3's SAE Handshake*. Rapport technique 383. 2019, p. 16. url : https://eprint.iacr.org/2019/383 (cf. p. 28).

[VT14] A. Varghese et D. Tandur. « Wireless requirements and challenges in Industry 4.0 ». In : *2014 International Conference on Contemporary Computing and Informatics (IC3I)*. 2014 International Conference on Contemporary Computing and Informatics (IC3I). Nov. 2014, p. 634-638. doi : 10.1109/IC3I.2014.7019732 (cf. p. 20).

[WD92] Christopher J. C. H. Watkins et Peter Dayan. « Q-learning ». In : *Machine Learning* 8.3 (mai 1992), p. 279-292. issn : 1573-0565. doi : 10.1007/BF00992698 (cf. p. 53, 55, 60, 75).

[Wei12] Yinxing Wei. *Federated Cross-Layer Access*. Internet-Draft draft-wei-abfab-fcla-02. Work in Progress, Expired & archived. Internet Engineering Task Force, mars 2012. 12 p. url : https://datatracker.ietf.org/doc/html/draft-wei-abfab-fcla-02 (cf. p. 42).

[WSJ17] M. Wollschlaeger, T. Sauter et J. Jasperneite. « The Future of Industrial Communication : Automation Networks in the Era of the Internet of Things and Industry 4.0 ». In : *IEEE Industrial Electronics Magazine* 11.1 (mars 2017), p. 17-27. issn : 1932-4529. doi : 10.1109/MIE.2017.2649104 (cf. p. 11, 20, 29).

[WW06] P. Wang et T. Wang. « Adaptive Routing for Sensor Networks using Reinforcement Learning ». In : *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*. Sept. 2006, p. 219-219. doi : 10.1109/CIT.2006.34 (cf. p. 59).

[WX17] X. Wu et L. Xie. « On the Wireless Extension of EtherCAT Networks ». In : *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. 2017 IEEE 42nd Conference on Local Computer Networks (LCN). Oct. 2017, p. 235-238. doi : 10.1109/LCN.2017.15 (cf. p. 22).

[Xia+09] B. Xia et al. « Reinforcement learning based spectrum-aware routing in multi-hop cognitive radio networks ». In : *2009 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications*. Juin 2009, p. 1-5. DOI : 10.1109/CROWNCOM.2009.5189189 (cf. p. 66, 67, 70).

[XP01] George Xylomenos et George C. Polyzos. « Quality of service support over multi-service wireless Internet links ». In : *Computer Networks* 37.5 (22 nov. 2001), p. 601-615. ISSN : 1389-1286. DOI : 10.1016/S1389-1286(01)00231-6. URL : http://www.sciencedirect.com/science/article/pii/S1389128601002316 (cf. p. 38).

[XS01] S. Xu et T. Saadawi. « Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks ? » In : *IEEE Communications Magazine* 39.6 (juin 2001), p. 130-137. ISSN : 0163-6804. DOI : 10.1109/35.925681 (cf. p. 27).

[Xu+16] L. Xu et al. « CogNet : A network management architecture featuring cognitive capabilities ». In : *2016 European Conference on Networks and Communications (EuCNC)*. 2016 European Conference on Networks and Communications (EuCNC). Juin 2016, p. 325-329. DOI : 10.1109/EuCNC.2016.7561056 (cf. p. 50, 51).

[Xu12] Xun Xu. « From cloud computing to cloud manufacturing ». In : *Robotics and Computer-Integrated Manufacturing* 28.1 (1er fév. 2012), p. 75-86. ISSN : 0736-5845. DOI : 10.1016/j.rcim.2011.07.002. URL : http://www.sciencedirect.com/science/article/pii/S0736584511000949 (cf. p. 16).

[Yi+11] Jiazi Yi et al. « Multipath optimized link state routing for mobile ad hoc networks ». In : *Ad Hoc Networks* 9.1 (1er jan. 2011), p. 28-47. ISSN : 1570-8705. DOI : 10.1016/j.adhoc.2010.04.007. URL : http://www.sciencedirect.com/science/article/pii/S1570870510000533 (cf. p. 34).

[YL94] Nathan Yee et Jean Linnartz. *Multi-Carrier CDMA in an Indoor Wireless Radio Channel*. Technical Report. USA : University of California at Berkeley, 1994 (cf. p. 33).

[Zim80] H. Zimmermann. « OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection ». In : *IEEE Transactions on Communications* 28.4 (avr. 1980), p. 425-432. ISSN : 0090-6778. DOI : 10.1109/TCOM.1980.1094702 (cf. p. 37).

[ZLZ15] K. Zhou, Taigang Liu et Lifeng Zhou. « Industry 4.0 : Towards future industrial opportunities and challenges ». In : *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Août 2015, p. 2147-2152. DOI : 10.1109/FSKD.2015.7382284 (cf. p. 15).

[Zor+15] M. Zorzi et al. « Cognition-Based Networks : A New Perspective on Network Optimization Using Learning and Distributed Intelligence ». In : *IEEE Access* 3 (2015), p. 1512-1530. ISSN : 2169-3536. DOI : 10.1109/ACCESS.2015.2471178 (cf. p. 44).

[Zur15] Richard ZURAWSKI, éd. *Industrial communication technology handbook*. Second edition. Industrial information technology series. OCLC : 935211357. Boca Raton London New York : CRC Press, Taylor & Francis Group, 2015. 1 p. ISBN : 978-1-138-07181-0 (cf. p. 20).

Table des figures

1.1 De la première révolution industrielle à l'Industrie 4.0, *extrait du site du Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)* 14

2.1 Structure d'une trame Ethernet extrait de *[Con18a]* 21
2.2 Structure d'une trame Ethernet contenant un télégramme EtherCAT, extrait d'ethercat.org 22
2.3 Structure d'une trame Ethernet contenant un télégramme Sercos III extrait de *[Con18b]* 23
2.4 Le rapprochement entre les technologies *IT* et *OT*, extrait de [WSJ17] 29

3.1 Représentations des différentes technologies réseaux en fonction de la fréquence et de leur portée. 35
3.2 Portion de la bande ISM 868 MHz utilisable pour des applications réseaux. 36
3.3 Les 6 différentes possibilités de conception inter-couches identifiées par A, B, C, D, E, F, extrait de [SM05]. 38
3.4 Architecture s'appuyant sur NeSt (*Network Status*) issue du projet MobileMAN, extrait de [BCD06] 40
3.5 Architecture d'OCARI, extrait de [Ber+08] 41
3.6 Architecture du *Cognitive Framework* extrait de [Tho+06] 45
3.7 Architecture de réseau cognitif avec prise de décision distribuée extrait de [Lee+07a] . 46
3.8 Architecture Cognitive networking framework extrait de [MRZ07a] 46
3.9 Architecture du CogNet AP extrait de [MRZ07a] 47
3.10 Résultats des tests de débit avec CogNet AP extrait de [MRZ07a] 48
3.11 Architecture Cognitive Complete Knowledge Network extrait de [MRZ08] 48
3.12 Architecture de *Cognet* extrait de [Dan+14]. 50
3.13 Architecture Cognet de Xu *et al.* extrait de [Xu+16] 51

4.1 Interaction entre l'agent et l'environnement dans un processus de décision Markovien, extrait de [SB18]. 54
4.2 Développement de l'apprentissage automatique extrait de [Bou+18]. Les algorithmes cités dans ce chapitre sont encadrés en vert. 55

4.3 Exemple de routes présentes sur le nœud 1 et à destination de 4 après exploration. . . 61
4.4 Grille irrégulière 6×6 de Boyan et Littman, extrait de [BL94]. 62
4.5 Délai de livraison en fonction de la charge réseau pour Q-routing et Bellman-Ford, extrait [BL94]. 62

5.1 Topologie de notre exemple jouet et placement des flux de données. Les chiffres sont les identifiants des nœuds. 79
5.2 Taux de paquets livrés en fonction du débit d'arrière-plan avec notre scénario jouet. . 80
5.3 Délai de livraison moyen en fonction du débit d'arrière-plan avec notre scénario jouet. 81
5.4 Gigue moyenne en fonction du débit d'arrière-plan avec notre scénario jouet. 82
5.5 La grille irrégulière 6×6 de Boyan and Littman [BL94] 82
5.6 Placement des 4 flux horizontaux. 83
5.7 Placement des 4 flux verticaux. 83
5.8 Chronogramme des flux de données sur la grille irrégulière. 83
5.9 Taux de paquets reçus pour l'ensemble des sources en fonction du débit sur la grille irrégulière. 84
5.10 Délai de livraison moyen pour l'ensemble des sources en fonction du débit sur la grille irrégulière. 85
5.11 Gigue moyenne pour l'ensemble des sources en fonction du débit sur la grille irrégulière. 86

6.1 Topologie de mise en évidence. Les nombres correspondent aux identifiants des nœuds. 89
6.2 Nombre de sauts moyen en fonction du temps simulé dans notre scénario de mise en évidence du caractère glouton de Q-routing. 90
6.3 Nombre de sauts moyen en fonction du temps simulé pour Q-routing avec mises à jour ε-gloutonne. 93
6.4 Nombre de sauts moyen en fonction du temps simulé pour Q-routing avec époques. Représentation boîte à moustache. 94
6.5 Taux de paquets livrés en fonction du débit d'arrière-plan avec notre scénario jouet sans-fil. 95
6.6 Délai de livraison moyen en fonction du débit d'arrière-plan avec notre scénario jouet sans-fil. 96
6.7 Gigue moyenne en fonction du débit d'arrière-plan avec notre scénario jouet sans-fil. . 97
6.8 Grille sans-fil adaptée de la grille de Boyan et Littman 98
6.9 Taux de paquets reçus pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic localisé). 99
6.10 Délai de livraison moyen pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic localisé). 100
6.11 Gigue moyenne pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic localisé). 101
6.12 Chronogramme des flux de données sur la grille irrégulière (trafic diffus). 101

6.13 Taux de paquets reçus pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic diffus). 102
6.14 Délai de livraison moyen pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic diffus). 103
6.15 Gigue moyenne pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil (trafic diffus). 104

7.1 Scénario avec mobilité utilisé pour mettre en évidence le problème d'estimation du délai. 106
7.2 Q-valeurs en fonction du temps simulé sur notre scénario avec mobilité. 107
7.3 Nombre de changements de route en fonction de la vitesse pour Q-routing. 108
7.4 Q-valeurs en fonction du temps simulé sur notre scénario avec mobilité avec Q-routing avec filtrage. 110
7.5 Nombre de changements de route en fonction de la vitesse pour les deux versions de Q-routing. 111
7.6 Taux de paquets reçus en fonction de la vitesse du nœud 9 112
7.7 Délai de livraison moyen en fonction de la vitesse du nœud 9 113
7.8 Gigue moyenne en fonction de la vitesse du nœud 9 114
7.9 Taux de paquets reçus pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil. 115
7.10 Délai de livraison moyen pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil. 116
7.11 Gigue moyenne pour l'ensemble des sources en fonction du débit d'arrière-plan avec la grille sans-fil. 117

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE NANTES

Titre : Réseaux cognitifs sans fil pour des applications industrielles 4.0

Mot clés : Réseaux cognitifs, Industrie 4.0, Q-routing, apprentissage par renforcement

Résumé : Dans le contexte de l'industrie du futur, les réseaux cognitifs sont une solution pour améliorer la fiabilité des réseaux informatiques et industriels. Ils sont capables d'optimiser automatiquement les différents paramètres protocolaires afin d'accomplir un ou plusieurs objectifs de qualité de service. Très peu de réseaux cognitifs ont été implémentés en totalité. La plupart des auteurs ont préféré se concentrer sur l'amélioration d'une fonctionnalité comme le routage. Dans cette thèse, nous suivons cette approche en proposant d'évaluer et d'améliorer l'algorithme Q-routing, algorithme de routage conçu par Boyan et Littman en 1994 et inspiré de Q-learning. Nous proposons une implémentation de Q-routing et des améliorations pour résoudre deux problèmes : les optimums locaux causés par la stratégie gloutonne de Q-routing et l'estimation du délai. Les optimums locaux privent Q-routing des routes ayant été congestionnées même momentanément. Pour résoudre ce problème, nous proposons deux approches inspirées des travaux sur l'apprentissage par renforcement. Par ailleurs, la qualité de l'estimation du délai est importante car elle est utilisée pour calculer la métrique de routage. Nous proposons d'utiliser une méthode de filtrage afin d'améliorer la qualité de l'estimation du délai de transmission. Nous évaluons notre implémentation et nos améliorations sur le simulateur réseau Qualnet incluant des topologies sans-fil et dans des scénarios avec mobilité. Nous montrons que Q-routing peut livrer plus de paquets et plus rapidement que le protocole de routage standardisé OLSRv2.

Title: Cognitive networks for Industrie 4.0 applications

Keywords: Cognitive networks, Industrie 4.0, Q-routing, reinforcement learning

Abstract: In the context of industry of the future, cognitive networks can help to increase robustness of computer and industrial networks. These networks are able to optimize automatically the different protocol parameters in order to perform one or more quality of service objectives. Unfortunately, cognitive networks have been rarely totally implemented. Most of the authors preferred improving only one functionality such as routing. In this PhD thesis, we follow this line by evaluating and improving Q-routing, a routing algorithm inspired by Q-learning and designed by Boyan and Littman in 1994. We propose an implementation of Q-routing and some improvements to solve two problems: local optimums due to the greedy strategy and the quality of delay measure. When a brief congestion happens on a route, this route can be never reused because of the greedy strategy. We propose two solutions inspired from reinforcement learning to solve local optimums problem. Otherwise, the quality of the estimation delay is also important. As Q-routing uses delay to compute routing metric, noisy measurements can let Q-routing doing the wrong choice. We propose to use filtering to improve the estimation of delay. We evaluate Q-routing and its modifications on discrete event network simulator Qualnet on several scenarios including wireless topologies and mobility. We show that our implementation can deliver more packets and faster than the standardized routing protocol OLSRv2.