



HAL
open science

Profiterole : un protocole de partage équitable de la bande passante dans les réseaux ad hoc

Rémi Vannier

► **To cite this version:**

Rémi Vannier. Profiterole : un protocole de partage équitable de la bande passante dans les réseaux ad hoc. Réseaux et télécommunications [cs.NI]. Ecole normale supérieure de lyon - ENS LYON, 2009. Français. NNT : 2009ENSL0547 . tel-00965326

HAL Id: tel-00965326

<https://theses.hal.science/tel-00965326v1>

Submitted on 25 Mar 2014

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.

Numéro d'ordre : 547

Numéro attribué par la bibliothèque : 147716 ENSL547

Profiterole

Un protocole de partage équitable de la bande passante dans les réseaux ad hoc

THÈSE

en vue d'obtenir le grade de

Docteur de l'Université de Lyon - École Normale Supérieure de Lyon
spécialité : Informatique
LABORATOIRE DE L'INFORMATIQUE DU PARALLÉLISME
École Doctorale de Mathématiques et Informatique Fondamentale

présentée et soutenue publiquement le 8 décembre 2009
à l'École Normale Supérieure de Lyon

par

Rémi VANNIER

Directeur de thèse : Isabelle GUÉRIN-LASSOUS

Professeur à l'Université Lyon 1

Après avis de :

Khaldoun AL AGHA
Fethi FILALI
David SIMPLOT-RYL

Professeur à l'Université Paris-Sud
Maître de conférences à EURECOM, Sophia-Antipolis
Professeur à l'Université Lille 1

Devant le jury :

Jean-Marie GORCE, Président du Jury
Fethi FILALI
Isabelle GUÉRIN-LASSOUS
Franck ROUSSEAU
David SIMPLOT-RYL

Professeur à l'INSA de Lyon

Maître de conférences à l'ENSIMAG, Grenoble

Mis en page avec la classe thloria.

Table des matières

Introduction	vii
I 802.11 dans les réseaux ad hoc	1
1 La norme 802.11	3
1.1 Réseaux ad hoc	3
1.2 Eléments de propagation radio	4
1.2.1 Modèles de propagation	4
1.2.2 Réception d'un signal	5
1.2.3 Imprécisions des modèles	6
1.3 La norme 802.11	6
1.3.1 Généralités sur IEEE 802.11	6
1.3.2 Couche Physique	7
1.3.3 Les architectures réseau de 802.11	7
1.4 Couche MAC & mode DCF	8
1.4.1 DCF	8
1.4.2 Un protocole de type CSMA/CA	8
1.5 Conclusion	12
2 Performances de 802.11 dans les réseaux multi-sauts	13
2.1 Débit utile de 802.11	13
2.1.1 Débit utile avec un seul émetteur.	13
2.1.2 Débit avec plusieurs émetteurs à portée.	14
2.2 Problèmes d'inégalité du partage dans les réseaux à un saut	15
2.3 Problèmes d'équité liés aux réseaux multi-sauts	16
2.3.1 Station cachée asymétrique	16
2.3.2 Les trois paires	17
2.3.3 Conclusion	18
3 Etat de l'art	19
3.1 Introduction	19
3.2 Solutions au niveau MAC	19

3.2.1	Solutions locales	20
3.2.2	Solutions exploitant le contenu des données	20
3.2.3	Solutions basées sur des paquets de signalisation	20
3.2.4	Transmissions forcées	21
3.2.5	DiffQ	21
3.3	Méthode par contrôle de débit	22
3.3.1	Modèles de contention	22
3.3.2	Approches par graphe de conflit	22
3.4	BRUIT	23
3.5	TCP dans les réseaux ad hoc multi-sauts	23
II	<i>Profiterole</i> : Proportionnal Fair Rate Control Protocol	25
4	Modèles et outils théoriques	27
4.1	Modélisation des interactions dans les réseaux 802.11	27
4.1.1	Définitions	28
4.1.2	Contraintes sur les ensembles indépendants maximaux	28
4.1.3	Contraintes par lignes	29
4.1.4	Contraintes par cliques	29
4.1.5	Graphes parfaits	30
4.2	Unit Disk Graphs	32
4.2.1	Introduction	32
4.2.2	Définitions et propriétés	32
4.2.3	Borne sur le degré d'imperfection	33
4.2.4	Conclusion	34
4.3	Voisinage à k sauts	34
4.4	Equité et Fonctions objectif	35
4.4.1	Définitions et notations	37
4.4.2	Maximisation de l'utilisation du réseau	37
4.4.3	Equité max-min	38
4.4.4	Equité proportionnelle	38
4.4.5	Synthèse des types d'allocation	38
4.5	Problèmes d'optimisation et relaxation Lagrangienne	39
4.5.1	Problèmes d'optimisation	39
4.5.2	Relaxation Lagrangienne	39
4.5.3	Généralisation à des contraintes multiples et par inéquations	40
4.5.4	Problème dual	41
4.6	Conclusion	42

5	Description de <i>Profiterole</i>	43
5.1	Solution théorique	43
5.1.1	<i>Profiterole</i> , un protocole <i>a la mode</i>	43
5.1.2	Enoncé du problème d'optimisation	44
5.1.3	Exemple	45
5.1.4	Utilisation de la relaxation Lagrangienne	46
5.1.5	Sous-problème 1 :	46
5.1.6	Sous-problème 2 :	47
5.2	Vers un algorithme distribué	47
5.2.1	Méthode de la plus grande pente	47
5.2.2	Convergence	47
5.2.3	Preuves de convergence avec erreur.	48
5.2.4	Algorithme de minimisation de \mathcal{L}	48
5.3	Implémentation pratique	49
5.3.1	Le protocole <i>Profiterole</i>	49
5.3.2	Contrôle de flux par token bucket	50
5.3.3	Calcul du surcoût du protocole <i>Profiterole</i>	51
III	Evaluation de <i>Profiterole</i>	53
6	Simulations	55
6.1	Le simulateur NS-2	55
6.1.1	Performances et réalisme	55
6.1.2	Inexactitudes du modèle de propagation radio de NS-2.33	55
6.1.3	Paramètres de simulation	56
6.2	Etude d'un flux	56
6.2.1	Flux CBR	56
6.2.2	Flux TCP	58
6.2.3	Flux à nombreux sauts	58
6.3	Station cachée asymétrique	59
6.4	Performances en débit sur de grands réseaux	62
6.5	Comparaison avec un protocole de couche MAC	64
6.5.1	Trois paires	65
6.5.2	Chaîne de communication	66
6.5.3	Scenarios aléatoires	66
6.5.4	Flux UDP	66
6.5.5	Flux TCP	67
6.5.6	Conclusion	68

7	Equité et fonctions d'utilité	69
7.1	Préambule	69
7.2	Introduction	70
7.3	Etat de l'art - notre approche	70
7.3.1	Etat de l'art	70
7.3.2	Notre approche	71
7.4	La couche Utilité	72
7.4.1	Services et QoS	72
7.4.2	Définition des fonctions d'utilité	73
7.4.3	Discussion	75
7.5	Intérêt de notre méthode d'évaluation	75
7.5.1	Simulations avec 5, 10 et 15 flux video UDP	76
7.5.2	Simulations avec une combinaison de flux UDP/CBR et TCP/FTP	79
7.6	Conclusions	79
	Conclusions	81
	A Annexes	83
A.1	Preuve du théorème de convergence avec erreur	83
A.2	Implémentation du token bucket dans Profiterole	84
	Table des figures	85
	Bibliographie	87

Remerciements

On m'avait prévenu qu'un travail de thèse s'apparente à une traversée de l'Atlantique à la rame. Je n'avais alors pas réalisé à quel point ces deux épreuves sont similaires. La nécessité de préciser ses recherches, de les limiter à un cadre étroit de sorte à aller jusqu'au fond des choses peut conduire à un dangereux isolement intellectuel. Une des conséquences de cet isolement est l'émergence d'un certain pessimisme : à mesure que l'on maîtrise le sujet, les concepts devenant de plus en plus simples et évidents, on peut avoir une tendance à dévaloriser son travail.

De la même manière, je pense que le rameur au milieu de l'Atlantique, une fois que l'enthousiasme des débuts s'est émoussé, passe par des périodes de doute angoissantes durant lesquelles il se demande ce qu'il fait là. La futilité de son défi occulte alors la grandeur de son exploit. Heureusement, de temps en temps, un hélicoptère passe filmer cet exploit, et vient lui rappeler que d'autres le contemplent. Même si l'exploit n'est heureusement pas aussi dangereux, j'ai éprouvé un sentiment semblable au passage de cet hélicoptère chaque fois que j'ai raconté mon sujet de thèse à d'autres, me rappelant ainsi le chemin déjà fait, et, à mesure que l'on me posait des questions, l'intérêt de mes recherches. Mes remerciements vont donc d'abord à tous les curieux qui se sont intéressés à mes recherches. Ensuite, je tiens à remercier très chaleureusement Isabelle, ma directrice de thèse, pour ses encouragements, et surtout pour sa patience. Je crois que, bien au delà des compétences (qu'elle a, bien évidemment), ces qualités sont les plus importantes pour encadrer une thèse. Et puis je voudrais remercier une deuxième Isabelle, ma femme¹, pour son soutien indéfectible, et pour les deux enfants qu'elle m'a donnés.

Merci encore à tous mes amis, et à tous ceux qui m'ont donné leur confiance, mes hélicoptères...

¹il s'agit bien de deux personnes différentes, je n'ose pas imaginer ce que cette thèse aurait été si elles n'avaient été qu'une.

Introduction

Les hommes sont mobiles, pas les réseaux.
Matthew S. Gast [31]

Cette remarque, issue de la bible de 802.11 de Matthew Gast, explique à elle seule le succès des technologies sans fil durant les vingt dernières années : ces technologies permettent un accès ubiquitaire aux données et aux réseaux qui sont, eux, intrinsèquement immobiles. En plus de la liberté qu'elles offrent aux utilisateurs de ces réseaux, les réseaux reposant sur ces technologies ont de nombreux avantages pour ceux qui les déploient : flexibilité, rapidité de déploiement et faible coût permettent de nombreuses applications, et soulèvent autant de défis qui n'ont pas encore été résolus par la communauté scientifique. Que ce soit pour interconnecter des sauveteurs sur une zone sinistrée, des militaires sur le champ de bataille, des capteurs d'humidité dans une forêt ou les différents appareils électroménagers de votre maison, l'impossibilité d'installer un réseau de câbles et le besoin d'un déploiement rapide rendent les technologies sans fil incontournables. Toutes ces applications rentrent dans le domaine des réseaux *ad hoc*, c'est-à-dire des réseaux sans fil autonomes et auto-organisés.

Les réseaux *ad hoc* ont ceci de différent des réseaux sans fils actuels, i.e. des réseaux avec point d'accès, qu'ils nécessitent encore moins d'infrastructure, puisque les entités qui les composent, plutôt que de communiquer exclusivement avec le point d'accès, sont des nœuds mobiles qui s'auto-organisent pour former un réseau. Du point de vue du concepteur de protocoles, cette topologie de réseau a deux implications : le fait que le réseau s'auto-organise impose de concevoir des mécanismes distribués, et la mobilité intrinsèque à ces réseaux implique que ces protocoles doivent être réactifs.

C'est à ces réseaux que je me suis intéressé pendant ma thèse, et plus spécifiquement aux difficultés qu'y pose la norme IEEE 802.11, norme des réseaux sans fil plus connue sous le nom de WiFi. On observe que dans ces réseaux, 802.11 partage souvent la bande passante de manière très inégale, avec les conséquences que cela entraîne sur les services utilisant ces réseaux, mais aussi sur la stabilité des liens radio. La couche MAC de 802.11, initialement développée pour les réseaux avec points d'accès, est à l'origine de ces problèmes.

On trouve deux types de solutions pour résoudre ces problèmes d'inégalités. La première solution consiste à proposer une alternative plus équitable à la couche MAC de la norme 802.11. Bien que de telles alternatives ont été proposées, ce type de solutions peut mettre un certain temps à voir le jour, car proposer une nouvelle couche MAC suppose une démarche fastidieuse de standardisation et de fabrication de nouvelles cartes réseau. Une autre solution à plus court terme consiste à réguler les débits des flux du réseau *au-dessus* de 802.11, c'est à dire dans les couches supérieures du modèle OSI. Ici encore, quelques rares solutions existent, mais leur complexité les rend difficiles à appliquer en pratique.

Le résultat principal de mes recherches est une proposition de protocole, appelé *Profiterole*, dont l'objectif est de réduire les inégalités de partage de bande passante que l'on constate quand on utilise la norme 802.11 dans les réseaux *ad hoc* multi-sauts. Ce manuscrit est donc avant tout un exposé de cette proposition de protocole et des raisons qui m'ont conduit à cette approche.

Le protocole *Profiterole* est issu d'un modèle simple et original censé représenter la contention dans les réseaux 802.11. La modélisation des interactions entre stations dans les réseaux sans fil est une tâche ardue car à la différence des réseaux filaires où les transmissions se font par des câbles, et donc dans des milieux confinés et maîtrisables, les transmissions radio se font dans un milieu non borné, sur un médium

changeant et imprévisible. Aussi, de nombreuses solutions que propose la littérature utilisent des modèles complexes pour tenter d'approcher au mieux la réalité de la propagation radio. L'expérience montre cependant que complexité ne signifie pas réalisme. D'autre part, la nécessité d'implémenter une solution sous forme d'un protocole présente un dilemme : plus un modèle est complexe, et plus son implémentation sera difficile. Dans les réseaux ad hoc, la marge de manœuvre est faible. La mobilité et la variabilité du medium radio imposent l'utilisation de protocoles réactifs et donc simples. La nouveauté de l'approche de *Profiterole* réside donc essentiellement dans la volonté que j'ai eu de la rendre applicable en pratique.

Cette thèse est découpée en trois parties. La première partie du manuscrit présente la technologie 802.11, les problèmes que cette norme pose dans le cas des réseaux ad hoc, ainsi que les solutions apportées par la recherche (chapitre 1, 2 et 3). La deuxième partie (chapitres 4 et 5) est consacrée à *Profiterole* et aux études théoriques qui ont conduit à cette idée. Enfin, la troisième partie est consacrée à l'évaluation et l'analyse de ce protocole par simulation (chapitres 6 et 7). En particulier, le chapitre 7 apporte un regard nouveau sur les métriques de qualité de service dans les réseaux sans fil, en proposant une méthode d'évaluation des protocoles de ces réseaux selon la satisfaction des utilisateurs. Cette méthode entend montrer les avantages d'une distribution équitable de la bande passante comme le fait *Profiterole*.

Première partie

802.11 dans les réseaux ad hoc

Chapitre 1

La norme 802.11

1.1 Réseaux ad hoc

La technologie WiFi a indubitablement modifié notre manière d'utiliser les ordinateurs. L'accès à Internet n'est désormais plus réservé aux ordinateurs fixes et à quelques téléphones portables aux abonnements prohibitifs. Désormais, un ordinateur portable, ou un iPhone peut se connecter à un point d'accès sans fil qui est, lui, connecté en filaire à l'Internet. Cette architecture de réseau en étoile, dont le centre est occupé par le point d'accès, est appelée WLAN, pour Wireless Local Area Network.

Aujourd'hui, la communauté informatique cherche de nouvelles solutions utilisant encore moins d'infrastructure, en supprimant soit le point d'accès - on parle alors de réseau ad hoc - soit la connexion filaire du point d'accès à Internet - on parle alors de réseau mesh. Ces architectures permettent des applications nouvelles. Souvent, ces applications ont des contraintes ou des propriétés particulières, et font l'objet de recherches spécifiques. C'est le cas des réseaux de capteurs² et des VANETS³.

Les réseaux maillés (ou réseaux mesh) : Une première idée pour réduire la quantité d'infrastructure consiste à limiter le nombre de points d'accès ayant un accès filaire à l'Internet. Comme pour un WLAN, les utilisateurs se connectent directement au point d'accès le plus proche. La différence réside dans le fait que l'ensemble des points d'accès forme un réseau sans fil de plus haut niveau. Ce réseau est en principe statique. Les opérations de routage y sont donc facilitées, et on peut même supposer qu'on y utilise des antennes directionnelles, ce qui réduit les problèmes d'interférence. Le projet RoofNet (de Roof : toit), un réseau urbain d'antennes sur le toit des immeubles [3], est un exemple concret de réseau maillé.

Les réseaux *ad hoc* : un réseau ad hoc est un réseau sans infrastructure, i.e. sans point d'accès. On y distingue les réseaux ad hoc simple saut, et multi-sauts.

- Dans les réseaux ad hoc simple saut, toutes les stations sont à portée de communication et communiquent directement entre elles. C'est la topologie de réseau idéale pour connecter ensemble plusieurs ordinateurs pendant une réunion sans utiliser de point d'accès.
- Dans les réseaux ad hoc multi-sauts, toute station fait office de relais, i.e. de routeur - pour d'autres communications. Ainsi, deux stations hors de portée de communication peuvent communiquer grâce à une chaîne de stations-relais, par *sauts*. Certaines de ces stations peuvent disposer d'un accès à Internet, permettant ainsi aux autres d'y accéder. Comme les stations sont a priori mobiles, la topologie du réseau change en permanence. Ce type d'application implique donc que les liens entre les stations peuvent être de courte durée, et que les re-routages sont fréquents. D'autre part, l'aspect multi-sauts soulève d'autres difficultés qui sont le cœur de cette thèse.

Les réseaux de capteurs sont des réseaux de petits équipements sans fil clairsemés dans une zone que l'on souhaite étudier. A titre d'exemple, la Queensland University of Technology de Brisbane étudie

²Sensor networks en anglais

³Vehicular ad hoc networks

l'impact de l'urbanisation sur la faune du Brisbane Forest Park [13]. Des capteurs répartis dans la forêt enregistrent les cris des oiseaux dans le but de mesurer les variations de la population, ou de détecter des modifications de leur comportement. Comme les capteurs n'ont pas assez d'énergie pour analyser ces informations, les données sont collectées via un réseau WiFi et traitées par une machine plus puissante⁴. Dans la mesure où les capteurs ont une batterie et une durée de vie limitée, c'est l'énergie qui est le facteur limitant dans l'élaboration de protocoles pour ces réseaux, alors que la bande passante et, pour certaines applications, le délai sont de moindre importance.

Les VANETS sont des réseaux de véhicules sur un réseau routier. Le fait que les voitures se déplacent de manière rectiligne facilite le travail de routage. C'est la mobilité qui est la principale contrainte dans ces réseaux. Les applications de ces réseaux sont nombreuses : les véhicules allant dans un sens peuvent prévenir les utilisateurs roulant dans l'autre sens d'un événement qui les attend, accident ou ralentissement brutal ; on peut aussi concevoir des applications distribuées pour permettre à chacun d'emprunter les routes les moins embouteillées, etc.

Le protocole *Profiterole* est une solution assez générique visant à réduire les inégalités de partage de bande passante dans les réseaux 802.11 multi-sauts. On l'appliquera donc plutôt à des réseaux où la bande passante est un facteur important, comme les réseaux mesh et les réseaux ad hoc multi-sauts.

1.2 Eléments de propagation radio

Le medium radio est peu fiable. Ce manque de fiabilité est à l'origine de tous les maux des concepteurs de protocole des réseaux sans fil. Il est donc difficile de parler de la norme 802.11 sans aborder les mécanismes de propagation des ondes radio. En particulier, de nombreux travaux dans les réseaux sans fil reposent sur des hypothèses simplistes concernant les modèles de propagation radio, puis fondent des calculs précis sur ces hypothèses. Pour le protocole de partage de bande passante que j'ai élaboré, *Profiterole*, j'ai choisi de sacrifier la précision du modèle pour la simplicité de l'algorithme. Puisque les modèles radio sont de toute manière imprécis, à quoi bon développer des protocoles complexes qui calculent le débit exact auquel une station peut émettre ? Bien que je me sois peu intéressé aux modèles radio pendant ma thèse, il me semble utile de montrer brièvement les difficultés que l'on rencontre pour modéliser la propagation radio.

1.2.1 Modèles de propagation

En l'absence d'obstacles, et pour des antennes omnidirectionnelles, la puissance reçue à une distance d est répartie sur une sphère de rayon d . Le signal s'atténue donc selon le carré de la distance parcourue. Ainsi, en espace libre, pour une puissance d'émission P_t , la puissance reçue par unité de surface est donnée par l'équation de Friis :

$$P_r^{(os)}(d) = P_t \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4\pi d} \right)^2$$

où G_t et G_r sont les gains des antennes émettrices et réceptrices, P_t et P_r^{os} ⁵ sont respectivement les puissances émise et reçue, et λ la longueur d'onde du signal. Cependant, dans un environnement réel, un signal se propageant dans toutes les directions va se réfléchir contre différents obstacles, comme des bâtiments, ou le sol. Chaque instance du signal va parcourir des distances différentes, et le signal reçu est donc la somme de plusieurs signaux déphasés. En fonction de la position du récepteur, ces interférences peuvent se cumuler, ou se détruire. Ce phénomène est appelé *multi-path fading*. Le modèle Two Ray Ground, utilisé dans le simulateur NS-2.33, consiste à modéliser l'impact d'une réflexion sur le sol. A

⁴Pour l'anecdote, les capteurs sont équipés de panneaux solaires qui paraît-il sont régulièrement obstrués par des reptiles qui apprécient la chaleur qu'ils dégagent, ce qui oblige les chercheurs à faire de temps en temps des expéditions assez désagréables.

⁵os : *Open Space*

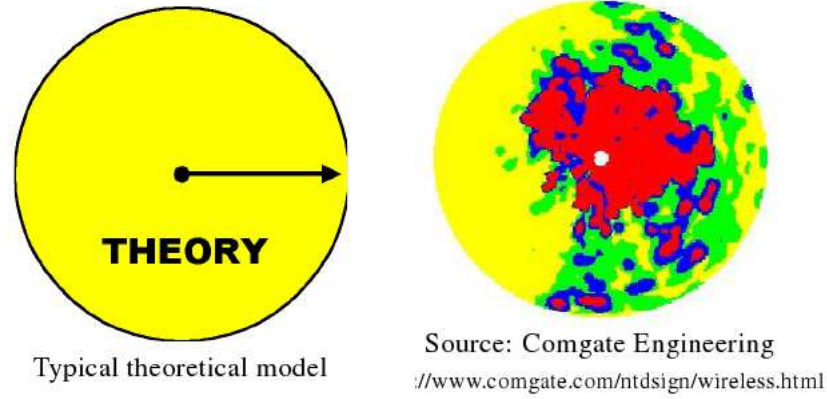


FIG. 1.1 – Carte de probabilité de réception d’un paquet envoyé par une station située au centre du cercle. À gauche, la théorie : la portée de l’émetteur est uniforme, et la probabilité d’erreur est la même en tout point. À droite, les performances d’une carte réelle.

faible distance, l’impact du multi-path fading est négligeable, et l’atténuation du signal suit le modèle en espace libre. Au-delà d’une distance seuil d_θ , définie par continuité, l’atténuation suit la loi :

$$P_r^{(2rg)}(d) = G_t \cdot G_r \cdot \frac{h_t^2 \cdot h_r^2}{d^4} \cdot P_t$$

où h_t et h_r sont les hauteurs respectives de l’émetteur et du récepteur. La distance d_θ est obtenue en résolvant $P_r^{(os)}(d) = P_r^{(2rg)}(d)$ ⁶.

De manière générale, les modèles de propagation radio utilisés dans la littérature sont de la forme :

$$P_r(d) = \frac{P_t}{d^\alpha}$$

où α est appelé constante d’atténuation. Elle vaut 2 en espace libre, entre 2 et 5 en environnement urbain, et varie de 1.6 à 6 en intérieur. La majorité des études théoriques dans les réseaux sans fil utilisent ce type de modèle.

Pourtant, ces modèles restent très imprécis, puisque la puissance reçue n’y dépend que de la distance à l’émetteur, indépendamment de l’environnement. Or, de nombreux facteurs peuvent affecter la puissance, tels que la présence d’obstacles, ou d’humidité. Et quand on sait qu’une simulation d’une centaine de stations pendant une centaine de secondes peut prendre plusieurs minutes avec NS-2, il semble difficile de modéliser d’autres phénomènes plus complexes comme la diffraction, l’absorption, l’effet Doppler, ou le fait que les antennes ne sont pas omnidirectionnelles. De manière générale, de nombreux travaux expérimentaux montrent que les modèles radios fondés sur la distance entre émetteur et récepteur sont très éloignés de la réalité [43, 5, 4]. Le contraste entre théorie et réalité est illustré par la figure 1.1.

1.2.2 Réception d’un signal

Afin de recevoir un signal, le récepteur doit distinguer le signal utile du bruit ambiant et des interférences causées dans la même bande de fréquence par d’autres émetteurs. Dit autrement, si la puissance du signal est trop faible par rapport aux interférences, celui-ci ne peut être décodé. Pour définir le seuil de puissance à partir duquel le signal peut être distingué des interférences, on utilise la notion de SINR (Signal to Interference plus Noise Ratio)⁷, exprimé en *dB* :

$$\text{SINR} = \frac{P_r}{P_i + P_n} \quad (1.1)$$

⁶ *2rg* fait référence au modèle *Two Ray Ground*

⁷ ou SNR - Signal to Noise Ratio - Rapport Signal sur Bruit en français

où :

- P_r est la puissance reçue,
- P_i est la somme des puissances des signaux interférents,
- P_n est la puissance du bruit ambiant, généré par l'agitation des particules dans l'air.

Si le SINR est élevé, le message peut être décodé, et plus il est faible, et plus grande est la probabilité d'erreur de transmission. Plus précisément, on considère que la probabilité pour qu'un bit de la trame soit corrompu, appelé BER (Bit Error Rate) est corrélée au SINR par une équation de la forme :

$$\text{BER} = \alpha \cdot \text{SINR}^{-\beta}$$

où α et β dépendent de la couche physique utilisée, ainsi que des performances des cartes sans fil⁸. Dans la plupart des simulateurs de réseaux sans fil, c'est un modèle à seuil qui est utilisé : si le SINR est supérieur à un certain seuil, tous les paquets sont reçus correctement. Sinon, aucun paquet ne peut être reçu. La version originale du simulateur NS-2.33 utilise un modèle de ce type, mais l'extension `dei80211mr`[24] permet d'y adjoindre un modèle à base de BER.

1.2.3 Imprécisions des modèles

Pour conclure sur cette section, voici six hypothèses sur la propagation radio identifiées par Kotz et al. [43] que l'on retrouve régulièrement dans la littérature et qui sont fausses en réalité :

1. le monde est plat,
2. la portée de transmission d'une antenne est isotrope : même les antennes omnidirectionnelles n'émettent pas avec la même puissance dans toutes les directions (voir, par exemple, l'introduction de la thèse de Claude Chaudet [16]),
3. les antennes ont toutes la même portée,
4. si je t'entends, tu peux m'entendre,
5. si je t'entends, alors je t'entends parfaitement,
6. la puissance reçue est une fonction simple de la distance entre émetteur et récepteur.

Il est important, lors de l'élaboration d'un protocole, de garder en tête ces imprécisions car un protocole qui repose de manière trop importante sur ces hypothèses pourra très probablement fonctionner sur la plupart des simulateurs de réseaux sans fil mais cessera de fonctionner sur une plateforme réelle. En particulier, dans la version 2.33 du simulateur NS-2 utilisé pour les simulations présentées dans cette thèse, seule la première hypothèse peut être rendue fausse. Les autres hypothèses sont toujours vraies.

1.3 La norme 802.11

1.3.1 Généralités sur IEEE 802.11

Le standard IEEE 802.11 [2] fait partie des nombreux standards de protocoles de communications proposés par le groupe IEEE (Institute of Electronics and Electrical Engineers). Leurs standards sont divisés en *projets*, dont le projet 802, dédié aux réseaux locaux (LAN, Local Area Network), est sans doute le plus célèbre. Chaque projet est lui-même divisé en groupes de travail : à titre d'exemple, IEEE 802.3 définit les couches physiques et MAC des réseaux à liaisons filaires, plus connu sous le nom d'*Ethernet*. Le standard IEEE 802.11, lui, décrit les couches basses du modèle OSI pour les réseaux sans fil :

- La couche PHY (physique), chargée de la transmission effective des données.
- La couche MAC (Medium Access Control).

Enfin, à l'intérieur du standard 802.11, plusieurs sous-groupes travaillent sur différentes facettes de la norme. Ceux-ci sont identifiés par une lettre apposée à 802.11. Quelques uns de ces sous-groupes ont produit des amendements à la norme, tels que 802.11a et 802.11b, qui spécifient deux couches physiques,

⁸pour des résultats expérimentaux, voir, par exemple, la probabilité de réception d'une carte Intersil HFA3861b dans [25].

Standard	Année	Bandes de fréquence	Débits physiques maximum	Modulation
Première version	1997	2.4-2.5 GHz	2 Mbit/s	DSSS
802.11a	1999	5.15-5.35 / 5.47-5.725 / 5.725-5.875 GHz	54 Mbit/s	OFDM
802.11b	1999	2.4-2.5 GHz	11 Mbit/s	HR-DSSS
802.11g	2003	2.4-2.5 GHz	54 Mbit/s	OFDM
802.11n	2007	2.4 GHz ou 5 GHz	540 Mbit/s	OFDM

FIG. 1.2 – Evolution des débits au cours des différents amendements de 802.11

l'une dans la bande des 5 GHz avec des débits allant jusqu'à 54 Mb/s, et la seconde dans la bande des 2.4 GHz, avec des débits allant jusqu'à 11 Mb/s. Aujourd'hui, la plupart des cartes sans fil utilisent la couche physique de l'amendement 802.11g, qui porte à 54 Mb/s les communications dans la bande des 2.4 GHz. Enfin, d'autres amendements portent sur la couche MAC, comme 802.11e, qui permet d'introduire de la qualité de service, ou 802.11i qui apporte des améliorations de sécurité.

1.3.2 Couche Physique

La couche physique est chargée de la transmission effective des données d'une station à une autre. Depuis sa création en 1997, le groupe de travail IEEE 802.11 a proposé de nombreuses couches physiques, variant par leur portée et leur débit. Les différents amendements qui ont apporté de nouvelles couches physiques sont donnés en table 1.2.

Nous parlerons peu de la couche physique dans la suite. Je me borne donc à donner les quelques éléments dont il sera question plus loin :

- Les différentes couches physiques de 802.11 ont un surcoût assez important. L'envoi d'une trame commence par un en-tête physique et un préambule PLCP (Physical Layer Convergence Procedure). Ce préambule permet de synchroniser émetteur et récepteur.
- Chaque couche physique permet d'émettre à une variété de débits physiques (par exemple, pour 802.11b, les débits 1, 2, 5.5 et 11 Mbps). En émettant à un débit faible, on réduit les risques de collision, et on augmente la portée de communication.
- La norme 802.11 stipule que les paquets de contrôle de la couche MAC doivent être envoyés aux débits physiques *de base*, c'est-à-dire les débits les plus faibles. Cette règle permet d'assurer une compatibilité avec les normes plus anciennes. Les débits physiques de base de 802.11b, par exemple, sont de 1 et 2 Mbps. Les paquets envoyés en broadcast sont eux aussi soumis à cette règle. Cette variété de débits est à l'origine du phénomène de *gray zone* : comme les paquets de contrôle et les paquets envoyés en broadcast sont envoyés à un débit plus faible, ils ont une plus grande portée. Par exemple, dans le protocole de routage AODV, chaque station maintient une liste de ses voisins à l'aide de paquets Hello, envoyés en broadcast. Ainsi, des stations qui sont hors de portée de communication apparaissent dans les listes de voisins, ce qui aboutit à de mauvaises décisions de routage.

1.3.3 Les architectures réseau de 802.11

Deux architectures sont possibles dans le standard 802.11 :

- L'architecture *ad hoc* permet à toutes les stations à portée de communication de communiquer entre elles. Un tel réseau est autonome et ne nécessite aucune infrastructure. Il est aussi dénommé, dans le jargon 802.11, IBSS pour Independent Basic Service Set.
- L'architecture basée sur une infrastructure qui nécessite la présence de points d'accès. Les stations du réseau ne sont autorisées à communiquer qu'avec le point d'accès auquel elles sont associées. Les points d'accès sont en général des routeurs équipés d'une interface WiFi et d'une interface Ethernet, et sont connectés sur cette deuxième interface à l'Internet. Le réseau constitué d'un point d'accès et des mobiles se trouvant sous sa zone de couverture est appelé BSS pour Basic Service Set. La combinaison de différents BSSs et du système de distribution permet de construire un réseau étendu. On parle alors d'ESS (Extended Service Set).

1.4 Couche MAC & mode DCF

1.4.1 DCF

La couche *liaison de données* est la seconde couche du modèle OSI, située entre couche physique (couche 1) et couche réseau (couche 3). Elle est divisée en deux sous-couches, la couche MAC (Medium Access Control), dont le rôle est d'arbitrer l'accès au médium entre plusieurs entités, et la couche LLC (Logical Link Control), qui permet de fiabiliser la couche MAC en introduisant des mécanismes de contrôle d'erreur et de contrôle de flux. La norme 802.11 ne contient que la partie MAC de la couche 2 du modèle OSI. C'est au niveau de la couche MAC que se trouve le cœur des problèmes d'inégalités dans le partage de bande passante dans les réseaux 802.11. C'est donc à cette couche que nous allons nous intéresser.

Il existe plusieurs modes de fonctionnement pour la couche MAC de 802.11, appelés *fonctions de coordination*. A l'heure actuelle, il en existe trois :

- Le mode DCF (Distributed Coordination Function) est le mode standard utilisé pour la plupart des réseaux sans fil actuels. Avant d'émettre, les stations vérifient que le médium est libre et, pour éviter les collisions, attendent un temps aléatoire appelé back-off.
- Le mode PCF (Point Coordination Function) permet une utilisation du médium sans contention. La coordination est assurée par un arbitre (typiquement, un point d'accès).
- Le mode HCF (Hybrid Coordination Function) est une invention plus récente, utilisée dans 802.11e et permettant d'introduire de la qualité de service dans les réseaux sans fil.

DCF est la fonction de coordination de prédilection des réseaux ad hoc, dans la mesure où elle est distribuée entre tous les nœuds du réseau, à la différence de *PCF*, plus adaptée aux réseaux avec points d'accès. D'autre part, la plupart des cartes sans fil et des points d'accès du marché fonctionnent sous *DCF*. C'est à ce mode que nous allons nous intéresser dans la suite.

1.4.2 Un protocole de type CSMA/CA

La principale difficulté que l'on rencontre dans les réseaux sans fil est le manque de fiabilité du médium. Alors qu'un fil de cuivre peut être dédié à des communications, les bandes de fréquence des communications radio peuvent être utilisées par d'autres équipements et brouiller les transmissions. Dans le cas de 802.11, ces bandes de fréquence sont les bandes ISM⁹, dans lesquelles émettent les fours à micro-ondes. En plus des interférences provenant d'autres appareils, la qualité du lien radio est imprévisible, autant à court terme (passage d'un obstacle, auto-interférences destructrices), que sur de plus longues périodes (variations d'humidité de l'air, par exemple).

La couche MAC de 802.11 en mode DCF est calquée sur le fonctionnement de la norme 802.3 pour les communications filaires (i.e. Ethernet) dans le sens que c'est un protocole de type CSMA. Cet acronyme signifie qu'une station doit écouter sur le médium et s'assurer que celui-ci est libre avant d'émettre afin d'éviter les collisions. Cependant, 802.11 DCF diffère sur deux points : l'utilisation de CSMA/CA à la place de CSMA/CD, et l'utilisation d'acquiescements positifs. Cet aspect de la couche MAC est en grande partie responsable des problèmes d'inégalité de partage du temps d'accès au médium.

Dans un contexte sans fil, il est difficile et parfois impossible pour un émetteur de détecter qu'une collision a eu lieu ou va avoir lieu. Pour les réseaux filaires, l'écoute du médium garantit que le médium est libre, du moins à l'instant de l'écoute. Dans le cas des transmissions radio, seul le récepteur, en mesurant le rapport signal à bruit, pourrait déterminer qu'une communication se passera correctement. Pour comprendre cette différence, il faut se rappeler que la puissance du signal est fortement atténuée entre émetteur et récepteur. Ainsi, un signal qui, du point de vue de l'émetteur, a une puissance négligeable par rapport au signal émis pourra, du point de vue du récepteur, avoir une puissance comparable avec le signal reçu. D'une part, l'émetteur peut donc difficilement savoir si la transmission se passera bien par la simple écoute du médium, et d'autre part, durant la transmission, l'émetteur est assourdi

⁹ISM : Industrial, Scientific & Medical

DIFS	50 μ s
SIFS	10 μ s
EIFS	364 μ s ^a
durée d'un slot	20 μ s
CW _{min}	31
CW _{max}	1023

^aCalcul effectué sur la base d'un acquittement envoyé à 1Mbps et un en-tête long (i.e. DSSS).

FIG. 1.3 – Temps d'attente dans 802.11b (source : [31])

par ses propres émissions et ne peut pas détecter une collision pendant qu'elle a lieu. Ainsi, plutôt que d'utiliser le mécanisme CSMA/CD d'Ethernet - Collision Detection - où une émission est immédiatement avortée quand une collision est détectée, 802.11 utilise le mécanisme CSMA/CA - Collision Avoidance, qui consiste à ajouter des temps d'attente aléatoires avant chaque émission pour réduire la probabilité de collision, et à augmenter exponentiellement ces temps d'attente si des collisions sont constatées.

La deuxième différence entre Ethernet et 802.11 provient de l'imprévisibilité du medium et de la fréquence des collisions : 802.11 prévoit l'utilisation d'*acquittements positifs*, c'est-à-dire que la station réceptrice envoie un acquittement pour chaque transmission réussie. Nous allons voir que ces deux différences ont un fort impact en termes de débit.

Fonctionnement de DCF

Le principe général de *DCF* est le suivant : avant d'envoyer une trame, une station doit attendre un temps fixe, appelé DIFS, puis un temps variable aléatoire appelé back-off. Si le medium est resté libre pendant ces temps d'attente, la station peut émettre. La partie aléatoire permet de réduire la probabilité que deux stations émettent simultanément. Ensuite, si la trame n'était pas une trame broadcast, le récepteur de la trame répond par un acquittement, envoyé après un temps d'attente plus court, appelé SIFS, et ceci indépendamment de l'état du medium.

Temps d'attente fixes. La norme 802.11 prévoit quatre temps d'attente fixes, dont seuls trois sont utilisés par DCF :

- SIFS (Short Inter-Frame Space), utilisé pour les acquittements. Celui-ci étant plus court que DIFS, les acquittements ont priorité sur les trames de données.
- DIFS (DCF Inter Frame Space) : sa valeur est fixée par la norme à $DIFS = SIFS + 2 \text{ slots}$, le slot étant l'unité de temps du protocole.
- EIFS (Extended Inter-Frame Space), utilisé en cas d'erreur de transmission. Ce temps est choisi pour permettre, en cas de collision avec une autre transmission, de laisser au récepteur de la transmission perturbatrice suffisamment de temps pour envoyer un acquittement. Sa valeur est elle aussi fixée par la norme à $EIFS = SIFS + DIFS + t_{ACK}$.
- PIFS (PCF Inter-Frame Space), qui n'est utilisé que dans le mode *PCF*.

Les valeurs de ces temps d'attente dépendent de la couche physique utilisée. A titre d'exemple, les valeurs de ces temps d'attente pour 802.11b sont résumées dans la table 1.3.

Temps d'attente aléatoire Comme tout émetteur doit attendre que le medium soit libre avant d'émettre, il est nécessaire d'ajouter des temps d'attente aléatoires pour limiter le risque que deux stations n'émettent simultanément. Ainsi, chaque station émettrice tire aléatoirement un nombre aléatoire b , appelé back-off, dans l'intervalle $[0..CW]$. Tant que le medium est libre, elle attend en décrémentant cette valeur de 1 à chaque slot. Si le medium est resté libre pendant b slots (i.e. la valeur du back-off atteint 0), la transmission peut commencer, et un nouveau back-off sera tiré à la prochaine émission. Dans le cas où le medium devient occupé pendant l'attente (une autre station a commencé à émettre), la station cesse de décrémenter son back-off. Quand le medium se libèrera de nouveau, elle attendra de nouveau DIFS, puis



FIG. 1.4 – Stations cachées : la station A souhaite émettre vers B . Une troisième station C est située à une distance telle que ses émissions ne peuvent être détectées par A par les mécanismes CSMA, mais sont susceptibles de provoquer des interférences en B . Dans l'exemple, C émet vers B , ce qui garantit qu'une collision aura lieu si A et C émettent simultanément mais ce cas de figure peut aussi se produire en choisissant pour C un autre récepteur que B (voir par exemple, le cas des stations cachées asymétriques). Ainsi, A et C sont susceptibles d'émettre simultanément, puisqu'ils ne peuvent s'entendre, et une collision se produit en B . On dit que C est une *station cachée* de A .

reprendra la décrémentation du back-off là où elle l'avait arrêtée. Comme son back-off au début de cette nouvelle phase de silence est statistiquement plus faible que le back-off d'une station qui vient d'émettre, celle-ci a plus de chances d'accéder au médium.

L'intervalle $[0..CW]$, appelé *fenêtre de contention*, peut être amené à varier en fonction de la disponibilité du médium. Si après une transmission, une station ne reçoit pas d'acquittement, elle suppose qu'une collision a eu lieu, et tente une nouvelle fois de transmettre en doublant la taille de sa fenêtre de contention. A chaque nouvelle erreur de transmission, la fenêtre de contention est doublée jusqu'à ce que la transmission réussisse ou qu'elle atteigne sa valeur maximale, au bout de 4 à 7 échecs consécutifs selon les variantes de 802.11. Dans le premier cas, la fenêtre de contention est remise à sa valeur initiale CW_{min} . Dans le second, quand la taille de la fenêtre de contention atteint CW_{max} , la transmission est abandonnée, et le lien considéré comme défectueux. Ce mécanisme de doublement de la fenêtre de contention, appelé *exponential backoff*, permet de réduire la probabilité de collision quand de nombreuses stations tentent d'émettre simultanément.

Seuil de détection de porteuse

Le *seuil de détection de porteuse* (Carrier Sense Threshold), exprimé en dB, est le niveau de bruit ou de signal maximum au-delà duquel une station considère que le médium est occupé, et qu'elle doit donc se restreindre d'émettre. Il est choisi par les constructeurs de cartes sans fil pour optimiser l'utilisation du médium. S'il est choisi trop élevé, l'absence de détection de porteuse ne garantit pas qu'il n'y aura pas de collision avec un autre signal (cas des stations cachées - voir figure 1.4). Inversement, s'il est trop bas, on réduit la réutilisation spatiale du médium puisqu'on interdit des communications qui pourraient avoir lieu (station exposée - voir figure 1.5).

Le vocabulaire utilisé pour la notion de distance de détection de porteuse porte à confusion : on trouve aussi le terme *distance d'interférence*. A cause de la manière dont sont modélisés les réseaux sans fil dans les simulateurs, il y a une confusion fréquente dans la littérature entre seuil d'interférence et seuil de détection de porteuse. Dans NS-2.31, une collision a lieu si deux conditions sont réunies 1) la station interférente est dans la zone de détection de porteuse du récepteur 2) le rapport entre puissance du signal utile et puissance du signal interférent est supérieur à un certain seuil. En pratique, la condition (1) a probablement été implémentée pour épargner au simulateur des calculs de SINR inutiles. Comme elle implique souvent la deuxième, on peut confondre les deux notions. Dans la suite de cette thèse, on dira que deux stations *s'entendent*, ou qu'elles sont à *distance de détection de porteuse* si les émissions de l'une dépassent le seuil de détection de porteuse de l'autre.

Le standard IEEE ne donne pas de recommandation pour le rapport optimal entre seuil de détection de porteuse et seuil de réception. Pour la couche physique DSSS, les seules informations disponibles dans le standard sont le taux d'erreur, qui doit être inférieur à 1% pour un signal reçu à -80dBm et un Clear Channel Assessment¹⁰ inférieur à -80dBm. Les constructeurs sont donc assez libres quant au choix du

¹⁰Clear Channel Assessment (CCA) : nom du signal envoyé par la couche physique à la couche MAC si le canal est libre

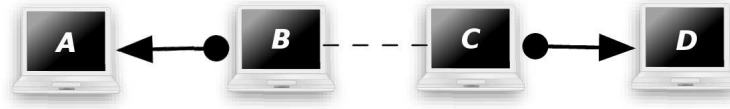


FIG. 1.5 – Le cas de la station exposée est le phénomène inverse. Deux stations B et C s’entendent et ont leurs récepteurs respectifs dans deux directions opposées, de sorte que si B et C émettaient simultanément, les deux communications se dérouleraient correctement, i.e. le rapport signal (P_B) sur bruit (P_C) au niveau du récepteur A est suffisant pour qu’il n’y ait pas collision.

seuil de détection de porteuse pour un fonctionnement optimal. C’est un choix complexe, qui a fait l’objet de nombreuses publications dans la littérature.

Selon une analyse théorique [74], le rapport idéal entre distance de communication et distance de détection de porteuse est compris entre 2.5 et 3.5. Ce rapport est obtenu en trois temps. Dans un premier temps, on calcule la densité maximale de stations *dans le plan*¹¹ de sorte que deux stations quelconques soient distantes d’au moins la distance de détection de porteuse, densité qu’on obtient en plaçant les stations sur les sommets d’un pavage triangulaire du plan. Ces stations forment donc un ensemble indépendant et peuvent toutes émettre simultanément. Dans un deuxième temps, on calcule le SINR dans le pire cas, obtenu pour une station réceptrice à la distance de communication. Enfin, le théorème de Shannon donne le débit maximal par lien que l’on peut obtenir avec un tel SINR. On obtient alors une valeur de débit par unité de surface en fonction du rapport entre distance de communication et distance de détection de porteuse. On cherche alors la valeur idéale de ce rapport selon la constante d’atténuation α du modèle de propagation radio (atténuation du signal en $d^{-\alpha}$). Pour un modèle de propagation en espace libre ($\alpha = 2$), le rapport idéal est autour de 2.5, alors que pour $\alpha = 4$, il est de 3.5. Il est à noter que ce rapport est indépendant du débit physique du protocole de communication utilisé, si l’on admet qu’il permet de transmettre à la capacité théorique prévue par le théorème de Shannon. Ainsi, ce rapport est autant valable pour 802.11b que pour les technologies plus actuelles.

Une autre approche théorique plus récente [75] confirme un rapport entre 2.5 et 3, approche qui a abouti à un protocole (réel, celui-ci) où les stations ajustent leur seuil de détection de porteuse de manière distribuée.

Les études théoriques de ce type sur le rapport entre distance de communication et distance de détection de porteuse ne manquent pas. Les études expérimentales, elles, sont rares. Une étude de Dominique Dhoutaut et al. [27] sur des cartes 802.11b montre que *“la distance d’interférence est au moins deux fois plus large que la distance de communication”*. Plus précisément, la distance d’interférence est ce qui est appelé distance de détection de porteuse dans cette thèse, et la distance de communication est celle obtenue avec un débit de 2Mbps, donc sensiblement plus grande que la distance de communication à 11Mbps.

Ce rapport va nous être utile dans le protocole Profiterole pour déterminer l’ensemble des stations susceptibles d’être à distance de détection de porteuse.

RTS/CTS

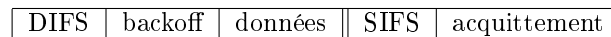
Les problèmes de stations cachées et de stations exposées sont dues au fait que c’est l’émetteur qui écoute le medium, alors que c’est au niveau du récepteur que le canal doit être libre pour éviter qu’une collision s’y produise. Malheureusement, pour connaître l’état du canal au niveau du récepteur, un échange de message est nécessaire. La norme IEEE 802.11 prévoit un tel mécanisme, appelé RTS/CTS. Cependant, compte tenu du coût de ces messages, ce mécanisme ne doit être utilisé que pour les paquets de grande taille.

Le principe est le suivant : avant d’émettre, une station émet un RTS (Request To Send), dans lequel

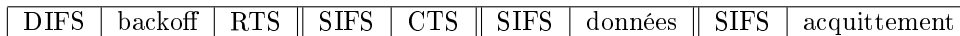
¹¹Notez qu’il s’agit d’une de ces hypothèses fausses en réalité citées plus tôt, en l’occurrence, l’hypothèse que le monde est plat

elle spécifie la durée de la transmission. Le destinataire répond par un CTS (Clear To Send) si le canal est effectivement libre de son côté. Les stations qui entendent le CTS ou le RTS, ou plus précisément, qui peuvent le décoder, sont ainsi prévenues qu'elles ne doivent pas émettre pendant la durée spécifiée. En pratique, chaque station maintient une structure de données appelée *Network Allocation Vector* dans laquelle sont mémorisées les plages de temps où elle ne peut émettre. Cette solution permet en principe de résoudre les problèmes de stations cachées, notamment dans les WLAN, où toutes les stations sont à portée de communication du point d'accès, et peuvent donc entendre un *RTS/CTS* envoyé par ce dernier. Le problème n'est en revanche pas résolu dans les réseaux ad-hoc multi-sauts, car les stations qui reçoivent le *CTS* ne sont pas tout à fait celles qui pourraient provoquer une collision. D'après [72], une grande majorité des stations susceptibles de provoquer des collisions ne peuvent pas décoder le *CTS*. En effet, selon les auteurs, *si une station peut recevoir le CTS, c'est qu'elle est à portée de communication du récepteur. Hors, la puissance nécessaire pour provoquer une collision est plus faible que celle requise pour recevoir correctement un paquet.* Ce raisonnement n'est peut-être pas tout à fait exact, car selon la norme 802.11, les *RTS/CTS* sont envoyés à un débit physique plus faible que les paquets de données. Ils sont donc reçus par des stations au-delà de la portée de communication des paquets de données.

Ainsi, les *RTS/CTS* ne résolvent pas complètement le problème des stations cachées. En pratique, il semble même peu recommandé de les activer dans le cas général. D'après Matthew Gast dans son livre à destination des administrateurs de réseaux 802.11 [31], la réduction du nombre de collisions compense rarement le coût des *RTS/CTS* en termes de débit. En effet, sans *RTS/CTS*, une communication atomique suit la séquence :



Avec les *RTS/CTS*, la séquence devient :



A titre d'exemple, dans 802.11b DSSS, les *RTS/CTS* ajoutent un total de $676\mu s$ ¹² à la durée d'une communication atomique, soit une augmentation de 45% pour l'envoi d'une trame de 1000 octets à 11Mbps ($1520\mu s$ tous temps d'attente confondus). Cependant, pour des trames longues, le coût des collisions peut devenir plus important que le coût de l'utilisation des *RTS/CTS*. Aussi, la norme 802.11 prévoit un seuil appelé $RTS_{\text{threshold}}$: les trames d'une longueur supérieure à ce seuil sont envoyées avec le mécanisme *RTS/CTS*.

1.5 Conclusion

Maintenant que les bases de 802.11 ont été posées, nous allons nous intéresser à ses performances dans les réseaux ad hoc multi-sauts. Le succès commercial de WiFi en a fait la technologie de référence des WLANs. C'est maintenant une technologie peu chère et bien connue. Naturellement, elle est donc devenue *de facto* la technologie sous-jacente des réseaux ad hoc. Nous allons voir cependant qu'elle pose de nombreuses difficultés quand on l'applique à ces réseaux.

¹² 2 SIFS ($2 * 10\mu s$), 2 préambules et en-têtes PLCP ($2 * 192\mu s$), RTS envoyé à 1Mbps ($160\mu s$), CTS ($112\mu s$)

Chapitre 2

Performances de 802.11 dans les réseaux multi-sauts

Comment 802.11 partage-t-il la capacité du médium entre plusieurs flux ? Les mécanismes de fenêtre de contention vus au chapitre précédent sont censés donner à chaque émetteur une chance égale d'accéder au médium. Ce mécanisme assure qu'un ensemble de stations en contention obtiendront un débit égal, mais cette égalité n'est pas toujours souhaitable en termes d'efficacité. D'autre part, les mécanismes de 802.11 sont nettement moins performants dans les réseaux multi-sauts, où l'on constate de sérieuses inégalités du partage de la capacité. Ces inégalités sont un point crucial des performances de 802.11. En effet, si un lien subit beaucoup de collisions ou obtient un débit nul, cela signifie que ce lien sera considéré par les couches supérieures comme un lien défectueux. Le routage dépend alors de saturations temporaires du médium radio, occasionnant ainsi des reconstructions de routes fréquentes, phénomène peu souhaitable dans un réseau.

L'objectif de ce chapitre est de présenter les performances de 802.11 en termes de débit et d'identifier les scénarios dans lesquels 802.11 alloue la bande passante de manière inégale ou inefficace afin d'en déduire les solutions qui résoudront ces problèmes.

2.1 Débit utile de 802.11

Quand une carte sans fil 802.11b/g affiche un débit maximal de $54Mbps$, il s'agit du débit *physique* de la carte, c'est-à-dire le débit qu'elle atteindrait si on supprimait les temps d'attente et d'acquittements de la couche MAC et les divers en-têtes des couches supérieures. Le débit *utile*, que l'on appellera désormais simplement "débit", est considérablement plus faible. A titre d'exemple, le débit maximal théorique d'une carte 802.11b/g seule est d'environ $30Mbps$ [31], et en pratique il est difficile d'atteindre plus de 20 ou $25Mbps$.

2.1.1 Débit utile avec un seul émetteur.

Je décris ici la durée d'une communication atomique dans 802.11b avec les couches physiques *DSSS* (en-têtes longs) et *HR/DSSS* (en-têtes courts), c'est-à-dire la transmission d'un paquet par l'émetteur puis de son acquittement MAC par le récepteur. Je rappelle que la couche *MAC* introduit deux temps d'attente fixes : un *DIFS* pour la transmission de la trame, et un *SIFS* pour l'acquittement. La couche physique introduit elle aussi des en-têtes (préambule et en-tête *PLCP*¹³), envoyés à des débits plus faibles pour assurer une compatibilité avec les équipements plus anciens. Pour la couche physique *HR/DSSS*, ces en-têtes sont transmis en partie à $1Mbps$ et à $2Mbps$, pour un total avoisinant les $100\mu s$. Ces nombreux en-têtes et temps d'attente réduisent considérablement le débit : une durée de $1324\mu s$ (respectivement $1516\mu s$ pour *DSSS*) est ainsi nécessaire pour transmettre un paquet de 1000 octets (voir tableau 2.1). On

¹³Physical Layer Convergence Procedure

	DSSS	HR/DSSS
DIFS	50 μ s	
Back-off moyen	310 μ s ^a	
Préambule	144 μ s	72 μ s
En-tête <i>PLCP</i>	48 μ s	24 μ s ^b
En-tête MAC (34 octets)		
Trame (1000 octets)	752 μ s ^c	
Total données	1304 μ s	1208 μ s
SIFS	10 μ s	
En-tête PLCP et préambule	192 μ s	96 μ s
Envoi d'un ACK (14 octets)	10 μ s ^{d,e}	
Total ACK	212 μ s	116 μ s
Total	1516 μ s	1324 μ s

FIG. 2.1 – Durée d’une communication atomique dans 802.11b *DSSS* en mode *DCF* (source : [31]). La version *HR/DSSS* avec en-têtes courts, plus récente, réduit la durée de la partie préambule et en tête *PLCP* à 96 μ s

^a 31/2 slots

^b 48 bits envoyés à 1Mbps (2Mbps)

^c $8 * 1034 / 1.1e7$

^d $8 * 14 / 1.1e7$

^e Certaines cartes envoient l’acquittement à un débit plus faible que 11Mbps

aboutit donc à un débit de 6.04Mbps (respectivement 5.28Mbps), au lieu des 11Mbps annoncés, soit une efficacité de 55% (respectivement 48%). A titre de comparaison, l’efficacité d’Ethernet 10M serait plutôt de 85%¹⁴ [11].

On peut tirer de ce paragraphe deux conclusions importantes :

- à cause des temps d’attente, dont la durée est incompressible, les trames courtes, tels que les acquittements de *TCP* ou les *RTS/CTS* de la couche MAC, occupent le medium pendant une durée significative.
- plus les débits sont élevés, et plus cette part incompressible devient importante. Certes, les variantes plus récentes de 802.11 ont été modifiés pour réduire cet impact - la taille maximale des paquets a été augmentée, et les temps d’attente légèrement réduits - mais ne compensent pas cette tendance, notamment car, pour des raisons de compatibilité avec les anciennes versions de 802.11, on ne peut réduire des temps d’attente comme les SIFS et les DIFS.

2.1.2 Débit avec plusieurs émetteurs à portée.

La capacité totale du medium radio varie aussi en fonction du nombre d’émetteurs qui émettent simultanément. La figure 2.2 montre le résultat d’une série d’expériences menées avec le simulateur NS-2, où l’on place un nombre croissant d’émetteurs sur une faible surface, de sorte que tous peuvent se détecter. La figure 2.2 représente l’évolution du débit total moyen en fonction du nombre d’émetteurs. Pour réduire la variance, chaque point est une moyenne de 10 expériences. Comme les émetteurs peuvent s’entendre, une collision ne peut se produire que dans le cas où deux stations finissent de décrémenter leur compteur de fenêtre de contention simultanément. Les collisions sont donc assez rares tant que le nombre d’émetteurs est faible.

Dans la toute première partie de la courbe (entre 1 et 3 émetteurs), la capacité totale augmente légèrement. Cela est dû au recouvrement des fenêtres de contention. En effet, même si un émetteur est seul, il doit attendre DIFS plus CW slots avant d’émettre, ce qui se traduit par un manque à gagner en termes d’utilisation du canal. Avec plusieurs émetteurs, ces temps d’attente se recouvrent. Statistiquement, le temps inter-paquet moyen devient, pour n émetteurs, DIFS + CW/ n slots. Au-delà de 4 émetteurs, en revanche, la capacité totale baisse. La présence croissante de collisions est à l’origine de cette baisse. En effet, quand une collision a lieu, chacune des stations ne se rendra compte qu’il y a eu collision qu’à la

¹⁴un seul émetteur, paquets de taille 1000 octets, le reste des paramètres étant choisis pour correspondre au mieux à une utilisation standard d’Ethernet. Avec deux émetteurs, l’efficacité est proche de 100%.

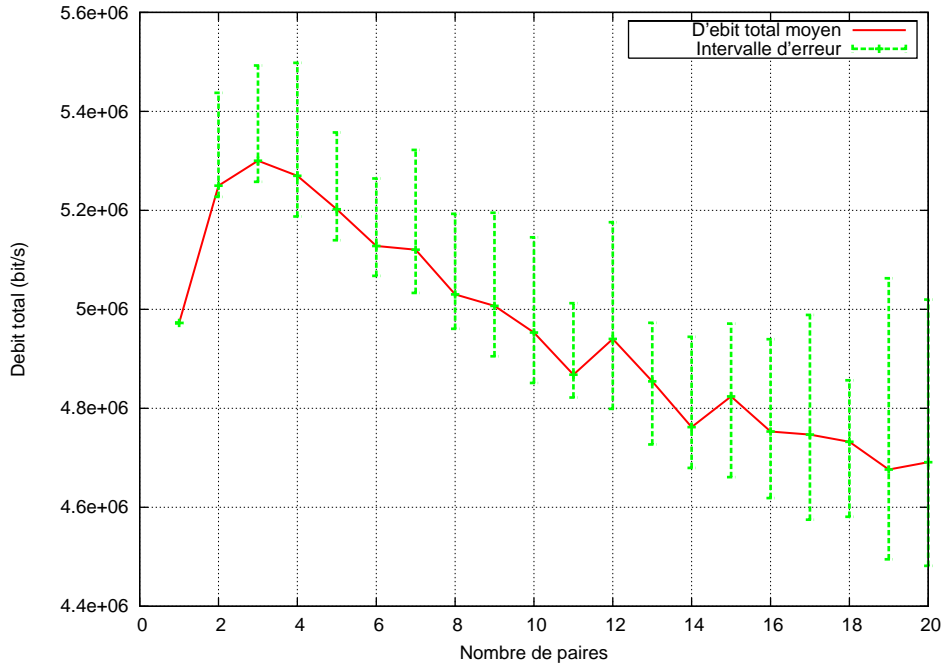


FIG. 2.2 – Débit total pour plusieurs émetteurs saturés dans une même zone de contention. Après une légère augmentation de capacité entre 1 et 3 émetteurs, le débit total se dégrade progressivement quand le nombre d'émetteurs augmente.

fin de l'émission, en l'absence d'acquittements MAC. Ainsi, la totalité de la durée d'émission des paquets corrompus est perdue. Après la collision, chacun des deux émetteurs va tirer un nouveau backoff dans une fenêtre de contention plus grande, ce qui aura aussi pour effet d'augmenter le temps inter-paquet moyen.

Cette caractéristique des protocoles CSMA/CA est à mettre en opposition avec la méthode de gestion des collisions de CSMA/CD, utilisée dans Ethernet, où les cartes réseau sont capables de détecter la collision pendant qu'elle a lieu et d'arrêter une transmission qui de toute façon ne sera pas reçue. Cette détection en cours de transmission est difficile pour des cartes sans fil, car seul le récepteur peut déterminer si la transmission s'est bien déroulée.

On retiendra de cette série d'expériences que tant que le nombre de stations en contention reste faible, la bande passante totale allouée à ces stations change peu. Cette constatation permet de justifier l'hypothèse de *divisibilité de la capacité* utilisée pour le modèle de partage de bande passante de *Profiterole* : si n émetteurs sont dans une même zone de contention, avec des débits respectifs x_1, x_2, \dots, x_n , alors l'occupation du médium sera sensiblement la même que si une seule station émettait à un débit de $\sum_{1 \leq i \leq n} x_i$.

2.2 Problèmes d'inégalité du partage dans les réseaux à un saut

Le mécanisme de back-off aléatoire donne *a priori* une probabilité égale d'accéder au médium à chaque émetteur et, sur le long terme, chaque station émettra statistiquement autant de paquets que les autres. Le fait qu'une station qui n'a pas pu accéder au médium conserve sa valeur de back-off et a donc plus de chances d'y accéder à la période de silence suivante tend à réduire les inégalités à court terme. Nous allons voir plus bas que le partage n'est pas toujours équitable quand les émetteurs ne sont pas dans la même zone de détection de porteuse. Mais même dans le cas où tous les nœuds sont dans la même zone, 802.11 peut présenter des anomalies, liées au fait que l'équité assurée par 802.11 n'est pas déterminée sur

des bases de bande passante, mais plutôt en termes de *nombre de paquets*. Par exemple, si un flux telnet et un flux *FTP* sont en contention, le premier sera nettement défavorisé en termes de bande passante. En effet, le protocole telnet, qui permet d'ouvrir un terminal sur une machine distante, utilise de nombreux paquets de petite taille alors que *FTP* envoie des paquets aussi gros que possibles (≈ 2300 octets pour 802.11b).

Le principe d'équité sur le nombre de paquets est aussi à l'origine de *l'anomalie de performance* [40]. On observe l'anomalie de performance quand plusieurs flux en contention ont des débits physiques différents. Comme le protocole 802.11 alloue à chaque flux le même nombre de trames, et que les trames envoyées avec un débit physique plus faibles prennent plus de temps, le medium est plus souvent - en proportion de temps - monopolisé par les flux au débit physique faible que par ceux ayant un bon débit physique. Paradoxalement, ce sont donc les flux les plus inefficaces auxquels 802.11 alloue le plus de temps.

La durée d'occupation du medium par une trame envoyée au débit physique ϕ avec des trames de longueur constante s , et un temps d'attente inter-paquets t_{ip} est :

$$t = t_{ip} + s/\phi \quad (2.1)$$

Le débit MAC utile agrégé d d'un ensemble de n flux de débit ϕ^+ s'obtient donc par l'équation suivante :

$$\frac{1}{d} = \frac{t_{ip}}{s} + \frac{1}{\phi^+} \quad (2.2)$$

Alors que celui d'un ensemble de $n - 1$ flux de débit physique ϕ^+ en contention avec un flux de débit physique ϕ^- est :

$$\frac{1}{d} = \frac{t_{ip}}{s} + \frac{n-1}{n\phi^+} + \frac{1}{n\phi^-} \quad (2.3)$$

On peut constater qu'au t_{ip} près, le débit agrégé d'un ensemble de n flux de débits physiques $\{\phi_1, \phi_2, \dots, \phi_n\}$ est la moyenne harmonique des débits qu'ils auraient seuls. Par exemple, si l'on met en compétition 11 flux à 11Mbps avec un seul flux à 1Mbps, leur débit agrégé sera approximativement celui d'un flux de débit physique 6Mbps. La seule présence de ce flux lent pénalise dramatiquement les flux rapides.

2.3 Problèmes d'équité liés aux réseaux multi-sauts

Dans un réseau où tous les nœuds sont à portée de détection de porteuse, les collisions sont rares puisqu'un nœud doit s'assurer que le medium est libre avant d'émettre. La notion de réseau ad hoc multi-sauts sous-entend que les distances entre certains nœuds d'un même réseau sont suffisamment grandes pour qu'ils ne se détectent pas. Dans ces réseaux, les collisions sont plus fréquentes et surtout touchent certains nœuds plus que d'autres. Ce type d'asymétrie où un nœud subit des collisions sans que le nœud perturbateur ne puisse en être informé est une des différences fondamentales entre les réseaux filaires et les réseaux sans fil. Dans un réseau filaire, les ressources partagées sont les câbles. Si un flux emprunte un câble ou traverse un routeur, il n'utilise que ce câble, ou ce routeur. Ainsi, en cas de compétition entre plusieurs flux, les routeurs disposent de toutes les informations nécessaires pour effectuer l'arbitrage. Dans le cas sans fil, une transmission a des conséquences sur des nœuds distants avec lesquels on ne peut pas nécessairement communiquer.

Les exemples suivants sont des situations pathologiques d'asymétrie dans le partage de la bande passante que l'on souhaitait résoudre avec le protocole Profiterole.

2.3.1 Station cachée asymétrique

Comme expliqué dans le chapitre 1, l'écoute du medium par l'émetteur ne garantit pas que le medium est libre au niveau du récepteur. Historiquement, le scénario de la station cachée fait référence au cas où deux stations A_1 et A_2 cherchent à communiquer avec un point d'accès mais sont trop éloignés pour s'entendre par les mécanismes de détection de porteuse. Quand les deux stations émettent simultanément,

une collision se produira vraisemblablement au niveau du point d'accès, aboutissant à la perte des deux messages. Comme la situation est symétrique pour les deux stations, des protocoles de contrôle de débit de type TCP peuvent résoudre partiellement cette difficulté : chacune des stations observe des pertes régulières de paquets et réduit donc son débit, réduisant ainsi les chances de collision, et aboutissant à un partage équitable du débit, certes assez peu efficace dans le cas de TCP en termes d'utilisation du médium.

Les réseaux ad hoc multi-sauts ont fait émerger d'autres problèmes de stations cachées qui peuvent, eux, être asymétriques : un flux subit peu de collisions, alors qu'il provoque sans le savoir des collisions sur le récepteur d'un autre flux. Dans l'exemple de la figure 2.3(a), l'obstacle situé entre A et C empêche ces deux stations de s'entendre. Ainsi, en l'absence de RTS/CTS, ces deux stations sont susceptibles d'émettre simultanément, auquel cas on observe des collisions au niveau de la station B . En revanche, le flux $C \rightarrow D$, lui, ne subit pas de collisions. Dans le cas où le flux $C \rightarrow D$ sature le médium, comme c'est le cas si c'est un flux TCP/FTP, par exemple, le flux $A \rightarrow B$ ne peut réussir une transmission que pendant les périodes de silence entre deux trames MAC. A moins que le flux $A \rightarrow B$ n'envoie des trames de faible taille, ces périodes ne sont pas assez longues pour qu'une telle transmission ait lieu, et celui-ci obtient un débit nul alors que le flux $C \rightarrow D$ obtient l'ensemble de la capacité.

Il est intéressant de noter que le mécanisme de RTS/CTS est insuffisant pour résoudre ce type de situation, alors qu'il l'est dans le cas des stations cachées classiques évoqué plus haut. En effet, même avec les RTS/CTS, le problème des stations cachées asymétriques apparaît si les stations B et C sont hors de portée de communication, auquel cas le CTS émis par B ne peut être décodé par la station C .

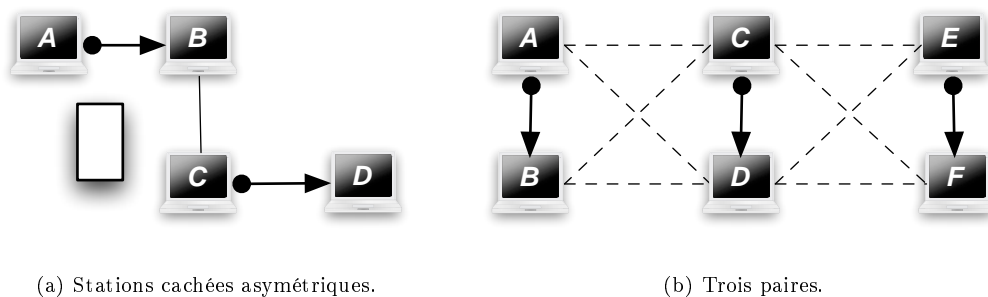


FIG. 2.3 – Deux scénarios problématiques des réseaux multi-sauts

2.3.2 Les trois paires

Le scénario des trois paires [26] que l'on retrouve aussi sous le nom "flow in the middle" [68] est un autre exemple d'asymétrie dans le partage de la bande passante, illustré en figure 2.3(b). Sur cette figure, deux stations sont reliées par des pointillés si celles-ci peuvent mutuellement s'entendre par les mécanismes de détection de porteuse. Les deux paires extrêmes, elles, ne s'entendent pas. Ainsi, quand la station A ou la station E émettent, la station C détecte leurs émissions et doit s'abstenir d'émettre. Inversement, les stations A et E peuvent émettre tant que C n'émet pas.

Une étude analytique de cette configuration [19] montre que si les paires extérieures ont toujours un paquet à émettre, la paire centrale ne peut émettre que très rarement (entre 2 et 5% du temps). Voici comment une telle asymétrie survient : on suppose que A commence à émettre. Puisqu'il entend le nœud A , C ne peut émettre et attend la fin de la transmission. En revanche, du point de vue de E , le canal est libre, et il peut donc émettre un paquet. Quand A termine sa transmission, seul E est en train d'émettre, et il peut donc commencer une nouvelle transmission... Tant que C n'émet pas, A et E sont indépendants et n'ont donc aucune raison de se synchroniser. Ainsi, C voit en permanence un canal occupé, sauf quand A et E sont simultanément dans une période d'attente. Si l'on suppose que C émet en premier, les stations A et E seront momentanément synchronisées, mais le problème évoqué ci-dessus finit par apparaître.

2.3.3 Conclusion

Dans cette section, je me suis attaché à montrer deux exemples d'inégalité du partage de la bande passante dans les réseaux ad hoc multi-sauts. Ils illustrent les deux types d'interaction de niveau MAC dans ces réseaux. Dans le scénario des trois paires, c'est la *contention*, c'est à dire le fait que deux stations qui peuvent se détecter ne doivent pas émettre simultanément, qui est à l'origine du problème. Dans le deuxième cas, il s'agit des collisions.

Dans les deux cas, un ou plusieurs émetteurs gênent une autre transmission sans le savoir. Ainsi, les stations perturbatrices n'ont aucune raison de réduire leur débit pour laisser de la place à la transmission perturbée. Le problème d'équité devient donc un problème d'information. Comment faire en sorte qu'une station qui perturbe d'autres stations en ait conscience? Quelles sont les informations pertinentes à extraire du réseau?

Chapitre 3

Etat de l'art

3.1 Introduction

Un certain nombre de problèmes liés à l'utilisation de 802.11 dans les réseaux ad hoc ont été mis en évidence dans le chapitre 2. Des solutions ont été depuis apportées afin de limiter l'apparition de ces problèmes et donc d'obtenir un meilleur partage de la bande passante du réseau entre les différents flux ainsi que de meilleurs débits.

Les solutions proposées peuvent être divisées en deux grandes classes. La première classe concerne les solutions au niveau MAC. En effet, les problèmes mis en évidence sont en partie dûs au protocole MAC de 802.11. Modifier ce protocole ou mettre au point un nouveau protocole MAC est donc une approche naturelle. Ces solutions se concentrent généralement sur un ou plusieurs problèmes identifiés et sont donc très efficaces sur les problèmes abordés. En revanche, elles sont souvent peu générales et ont du mal à régler tous les problèmes rencontrés avec 802.11. De plus, ces approches ont l'inconvénient, pour être utilisées en pratique, qu'elles doivent être intégrées à une technologie sans fil, ce qui nécessite très souvent une phase de standardisation, phase pas si simple et fastidieuse. La deuxième classe de solutions considère que la technologie sans fil 802.11 a encore de beaux jours devant elle, et qu'il y a peu de chances pour que son protocole MAC soit modifié de fond en comble. Par conséquent, ces solutions supposent que la technologie sous-jacente est 802.11 et se placent au-dessus de cette dernière. Dans cette classe, les solutions cherchent à réguler le débit entrant au niveau MAC afin de limiter l'apparition de congestion et de collisions.

Ce travail de thèse ne s'intéresse qu'aux réseaux sans fil multi-sauts ayant des stations équipées d'une seule carte sans fil et communiquant entre elles sur un seul canal. Bien des solutions ont été élaborées pour des systèmes multi-interfaces ou multi-canaux. Si ces approches permettent très souvent d'améliorer l'efficacité du système, elles n'enlèvent pas tous les problèmes mentionnés dans le chapitre 2 et qui apparaissent dans le sous-réseau fonctionnant sur un même canal. De plus, ce travail permet d'avoir une idée des performances qui peuvent être obtenues dans le cadre le plus contraint, i.e. dans un réseau mono-canal.

3.2 Solutions au niveau MAC

Les solutions proposées au niveau MAC basées sur un seul canal sont multiples. Elles cherchent assez souvent à répondre à un problème spécifique et connu avec 802.11. Ces solutions peuvent être dans la veine de 802.11 (i.e. le paramétrage ou les algorithmes de 802.11 sont simplement modifiés), ou ont une approche très différente. Elles peuvent être rangées selon trois classes principales :

- Les solutions locales utilisent essentiellement des informations locales. Une information locale correspond à une information connue par le nœud sans communication explicite supplémentaire avec ses voisins. Par exemple, un nœud sait si son paquet a été transmis avec succès ou non selon qu'il reçoit ou non le paquet d'acquittement de son paquet de données. Un nœud sait aussi s'il y a de

l'activité sur le canal radio grâce à son mécanisme de détection de porteuse. 802.11 appartient à ce type d'approches.

- Certaines solutions exploitent les informations contenues dans les paquets de données et de contrôle qui passent sur le réseau. Chaque nœud peut lire les informations se trouvant dans les paquets qui lui sont destinés. Il peut aussi, dans le mode promiscuous, lire les informations de tous les paquets qu'il peut décoder même si ceux-ci ne lui sont pas destinés. Une telle information permet d'inférer le taux d'utilisation du nœud ainsi que celui de ses voisins.
- Certaines solutions vont se baser sur des paquets de signalisation envoyés explicitement afin d'apporter de la connaissance aux nœuds voisins sur, par exemple, la topologie ou l'utilisation du réseau. L'utilisation des RTS et CTS dans 802.11 correspond à une telle approche. En général, on considère que les solutions de cette classe utilisent des paquets de signalisation supplémentaires autres que les classiques RTS/CTS de 802.11.

Comme vous le verrez par la suite, la solution proposée dans cette thèse n'est pas une solution MAC mais est basée au-dessus de 802.11. Par conséquent, l'état de l'art sur les solutions MAC donnent les idées générales et les principaux protocoles MAC qui ont été proposés afin de montrer leurs apports mais aussi leurs limitations. Le lecteur intéressé par cette approche peut se référer aux sections état de l'art des différents articles cités sur le sujet et/ou à des articles de synthèse sur le sujet, comme par exemple à [44].

3.2.1 Solutions locales

Comme expliqué précédemment, ces solutions ne se basent que sur des informations locales sans échange explicite d'informations pour prendre des décisions. Le mode DCF de 802.11 sans utilisation de RTS/CTS fait partie de cette classe. En effet, le mécanisme d'accès au médium de 802.11 dépend du résultat de l'écoute du canal (libre ou occupé) et du résultat sur le paquet précédemment émis (émis avec ou sans succès). Ces informations sont disponibles localement. Plusieurs protocoles d'accès au médium ont été proposés dans la veine de 802.11. Les principales modifications concernent les valeurs des paramètres (comme les valeurs des temps d'attente fixes par exemple) et/ou l'algorithme de gestion de la fenêtre de contention. Des algorithmes comme MILD (Multiplicative Increase Linear Decrease) [10], DIDD (Double Increase Double Decrease) [15] et SBA (Simple Backoff Algorithm) [60] font partie de cette classe puisqu'ils modifient l'algorithme de gestion de la fenêtre de contention. D'autres solutions comme PNAV [17] ou MadMac [61] ajoutent de nouveaux temps d'attente en fonction d'informations locales.

3.2.2 Solutions exploitant le contenu des données

Les solutions de cette classe ont un fonctionnement relativement similaire à 802.11. La principale différence réside dans le fait que les valeurs choisies pour les paramètres et/ou l'algorithme de gestion de la fenêtre de contention ne sont plus fonction d'informations locales mais d'informations extraites des paquets qui peuvent être décodés correctement. Les paquets utilisés ne sont pas nécessairement des paquets envoyés par cette station ou destinés à cette station. Ce peut être tout paquet transitant sur le médium radio et pouvant être décodé. La solution MBFAIR [7] fait partie de cette catégorie. Dans cette solution, chaque station détermine individuellement quel doit être le partage équitable du médium radio en fonction des paquets de données, RTS, CTS et acquittements qu'elle reçoit. Selon ses calculs, la station sait si elle utilise le médium plus ou moins que ses voisins. La station ajuste alors la taille de la fenêtre de contention en conséquence. Les protocoles [48, 54] font aussi partie de cette classe.

3.2.3 Solutions basées sur des paquets de signalisation

Dans cette classe, les solutions vont utiliser des informations autres que celles fournies par 802.11, notamment des paquets de signalisation supplémentaire. Le protocole EHATDMA [38] entre dans cette classe. Dans cette solution, un échange d'informations se fait entre l'émetteur et le récepteur, échange pouvant être initié par une des deux parties.

Toutes les solutions mentionnées précédemment respectent l'approche CSMA/CA et n'appliquent pas de gestion différenciée entre les différents paquets, i.e. chaque station a une seule file d'attente au niveau MAC et le même mécanisme est appliqué sur toutes les stations. Les deux sous-sections suivantes décrivent deux approches différentes. La première ne suit pas l'approche CSMA/CA tandis que la deuxième a un mécanisme d'accès différencié. Comme ces deux solutions originales cherchent à résoudre (ou réduire) certains problèmes soulevés par l'utilisation de 802.11 dans les réseaux sans fil multisautes, je les inclus dans cet état de l'art.

3.2.4 Transmissions forcées

On trouve une approche simple et astucieuse dans [55] dont l'objectif initial est de résoudre le problème des trois paires. Dans cette configuration, décrite dans le chapitre précédent, la paire centrale ne peut émettre car elle doit attendre que les deux paires extérieures cessent simultanément d'émettre. Pour résoudre cette difficulté, les auteurs de [55] proposent un mécanisme simple : si une station constate qu'elle ne parvient pas à accéder au medium, elle en déduit qu'elle se trouve dans une configuration où deux émetteurs ou plus sont à distance de détection de porteuse, mais ne s'entendent pas mutuellement. Dans ce cas, elle émet un paquet uniquement destiné à brouiller les transmissions des autres émetteurs, afin de leur signaler sa présence. Ces émetteurs, observant des collisions, augmentent alors la taille de leur fenêtre de contention, donnant ainsi plus de chances à la station initialement bloquée d'accéder au medium.

La simplicité est certes un argument de poids dans l'élaboration des protocoles. Provoquer des collisions est d'ailleurs le seul recours pour interagir avec une station hors de portée de communication. Mais on peut reprocher à cette solution deux inconvénients. D'une part, ce mécanisme a un coût en termes d'utilisation du medium, puisque la totalité de la durée d'occupation de la trame ayant subi la collision est perdue. D'autre part, ce mécanisme souffre des mêmes inconvénients que les RTS/CTS : la puissance nécessaire pour provoquer une collision n'est pas la même que celle pour détecter une porteuse.

3.2.5 DiffQ

DiffQ [68] est une autre solution originale visant à résoudre les problèmes d'équité au niveau MAC, s'inspirant des files à priorité de 802.11e. Le principe est le suivant. Chaque station maintient une file d'attente MAC pour chacune des destinations des flux qu'elle transmet. D'autre part, pour des flux multi-sautes, elle communique la taille de sa file d'attente MAC à la station la précédant sur le chemin du flux. La priorité des paquets est choisie en fonction de la différence entre la taille de la file du saut suivant, et celle de la file locale, sachant qu'on considère que la file d'attente au niveau de la destination du flux est d'une taille nulle. Les mécanismes de priorité utilisés sont les mêmes que dans la couche MAC de 802.11e, où les temps d'attente fixes (DIFS) et variables (taille de la fenêtre de contention) varient en fonction du type de paquet.

Ainsi, pour un flux multi-sautes passant par les stations X-Y-Z, si la qualité du lien Y-Z baisse, ou qu'il subit une forte contention, la file d'attente de la station Y grossit. Le mécanisme proposé donne donc une priorité plus grande à Y pour l'accès au medium, mécanisme qui tend à équilibrer les tailles des files d'attente des stations en contention.

De manière générale, ce problème résout assez bien les problèmes de 802.11 liés à la contention, tels que le problème des trois paires. En revanche, il fonctionne moins bien dans les problèmes liés aux collisions, comme le problème de la station cachée asymétrique. Dans ce cas, bien que DiffQ donne une priorité plus grande au flux qui subit les collisions, et donc des temps d'attente plus faibles, rien n'interdit que les deux stations émettent simultanément, puisqu'elles sont censées ne pas s'entendre. Ainsi, l'augmentation relative des temps d'attente du flux perturbateur ne réduit pas vraiment la probabilité de collision.

Une fois de plus, le fait qu'un flux puisse perturber un autre flux sans le savoir incite à choisir une solution pour résoudre ces problèmes à un niveau plus haut que le niveau MAC, à un niveau qui bénéficie

d'informations moins locales et plus pertinentes que la détection de porteuse, les pertes de paquets, ou les tailles des files d'attente.

Ce protocole sera étudié plus en détail dans la partie évaluation, où nous comparons les performances de Profiterole avec DiffQ dans une variété de situations.

3.3 Méthode par contrôle de débit

Les solutions de cette classe cherchent à réguler le débit entrant au niveau de la couche MAC. Une grande partie de ces solutions considère très souvent que la technologie sans fil sous-jacente utilisée est 802.11. La régulation de débit est effectuée via un mécanisme d'allocation des débits dans le réseau afin de limiter l'apparition de congestion et d'obtenir une meilleure équité entre les flux. Cette allocation repose sur un modèle qui cherche à exprimer la contention et le partage dans un réseau sans fil. A partir de ce modèle, un graphe de conflit est construit afin de déterminer les dépendances existant entre les différentes stations et/ou les différents liens radio du réseau. A partir de ce graphe de conflit, différents algorithmes d'allocation de débit ont été proposés. Ce chapitre propose un état de l'art sur les approches proposées. Le protocole *Profiterole* fait partie de cette classe de solutions. Je reviendrai donc sur les notions de graphe de conflit de manière plus formelle dans le chapitre 4.

3.3.1 Modèles de contention

Les deux principaux modèles qui représentent la contention dans le réseau, et qui sont très souvent repris dans la littérature, sont les modèles proposés par Gupta et Kumar [34]. Dans le *modèle physique*, une transmission est reçue avec succès si le rapport signal sur bruit au niveau du récepteur est supérieur à un certain seuil. Dans le *modèle protocolaire*, plus facile à appliquer en pratique, un récepteur peut recevoir avec succès les paquets envoyés par une source si ces deux nœuds sont à portée de communication et qu'aucun autre nœud est en train de transmettre dans la zone d'interférence du récepteur. Il est possible d'adapter ce modèle à 802.11 en y ajoutant la contrainte qu'aucun nœud ne doit transmettre dans la zone de détection de porteuse ni dans la zone d'interférence de l'émetteur.

Ces deux modèles de contention sont très souvent utilisés pour construire le graphe de conflit qui traduit les dépendances existant dans le réseau.

3.3.2 Approches par graphe de conflit

On trouve de nombreuses propositions de protocoles basées sur le modèle du graphe de conflit. Je cite deux exemples intéressants de la littérature, le premier reposant sur le modèle physique, le second sur le modèle protocolaire.

Dans [71], chaque paire de liens du réseau fait l'objet de calculs précis sur les niveaux de SINR de sorte à déterminer s'ils peuvent émettre simultanément. Ses auteurs supposent que la position géographique de chaque station est connue, par exemple en utilisant GPS, et calculent donc ces SINR en fonction de la distance qui les sépare. Une fois ce graphe de conflit "précis" établi, les auteurs proposent un algorithme de calcul des débits optimaux sur ce graphe de conflit. Les auteurs de [47] ont eux aussi une approche de ce type.

On ne peut pas nier la qualité théorique de cette approche. Il est cependant assez clair que la qualité de ce type de solutions dépend essentiellement de l'adéquation du modèle radio à la réalité. Par exemple, il est peu probable qu'un tel protocole soit utilisable dans des réseaux sans fil urbains, où (1) l'éloignement géographique d'un émetteur est un indicateur trop peu fiable de la puissance à laquelle le signal sera reçu et (2) les variations à court terme du SINR peuvent atteindre 20dBm d'une mesure à l'autre (voir notamment les résultats pratiques de [39]). La faiblesse de ce type d'approches est donc qu'on ne peut probablement les utiliser que dans de rares applications où le medium radio est fiable et bien connu.

Dans [73], les sommets du graphe de conflit sont les liens du réseau, et on considère que deux liens sont en contention si la source ou la destination de l'un des liens est à une distance inférieure à d_{int} de

la source ou de la destination de l'autre lien, suivant donc le modèle protocolaire. Pour déterminer les débits optimaux auquel chaque lien peut émettre, les auteurs de [73] utilisent un ensemble de contraintes appelées contraintes par cliques. Ces contraintes par cliques imposent que la somme des émissions sur toute clique du graphe de conflit doit être inférieure à la capacité du réseau. Je reviendrai dans le chapitre suivant sur ces concepts. Chaque lien actif calcule donc les cliques maximales du graphe de conflit dont il fait partie, puis le calcul des débits optimaux est fait de manière distribuée, en utilisant une méthode de "pricing".

Cette solution, élégante en théorie, souffre de deux inconvénients. Premièrement, si l'on fait abstraction des imprécisions du modèle protocolaire, l'allocation obtenue avec les contraintes par cliques est à peu de choses près une allocation optimale. Mais en plus de ces imprécisions, des approximations sont faites pour accélérer le calcul des cliques car celui-ci peut être complexe. D'autre part, l'algorithme de calcul des débits semble converger lentement : 10 secondes sur les topologies les plus simples, 100 secondes sur des topologies plus complexes (en l'occurrence, 30 stations). Pendant la période de convergence, les débits imposés au réseau peuvent donc être arbitraires. Cette lenteur de convergence est donc critique pour de tels protocoles, et il est essentiel d'en améliorer les propriétés, par exemple en simplifiant le modèle, élégant théoriquement, mais finalement assez peu pertinent au regard des approximations faites sur la modélisation radio.

3.4 BRUIT

BRUIT (Bandwidth Reservation Under InTerferences) [20] est une proposition de protocole permettant de faire de la réservation de ressources, c'est-à-dire de déterminer avant qu'un flux commence à émettre, s'il y a suffisamment de bande passante disponible pour ce flux, puis si c'est le cas, à informer les stations voisines de cette réservation, afin de donner des garanties de qualité de service aux flux. BRUIT est une approche privilégiant la simplicité du modèle de contention, et considère que deux stations sont susceptibles d'interférer si elles se trouvent à 2 sauts radio ou moins. Le principal intérêt de cette solution est l'introduction de la notion de sauts radio, qui permet notamment d'éviter l'utilisation de la distance comme facteur déterminant de la possibilité de contention.

3.5 TCP dans les réseaux ad hoc multi-sauts

Le rôle du protocole TCP est d'assurer un transport fiable des données de la source à la destination dans des réseaux qui ne sont pas fiables. En théorie, et selon le modèle OSI, TCP devrait être indépendant des couches inférieures, et en particulier du type de réseau dans lequel il est utilisé. En pratique, ce n'est pas le cas, car TCP a été développé sur des hypothèses liées aux réseaux filaires.

En particulier, un des rôles de TCP est d'éviter la congestion. Pour ce faire, TCP adapte le débit en fonction des pertes de paquets qu'il observe, perçues selon les versions de TCP sous forme d'acquittements dupliqués, ou d'expiration de timeout. Dans la version de base de TCP, tous les paquets sont acquittés par la destination, et l'acquittement contient le numéro du prochain paquet que la destination s'attend à recevoir. Si dans la séquence de paquets reçus, la destination constate qu'un paquet manque, elle continue d'envoyer des acquittements contenant le numéro de séquence du paquet manquant, d'où la dénomination "acquittements dupliqués". Si une perte de paquet est constatée, TCP réduit la taille de sa fenêtre de congestion, ce qui a pour effet de réduire le débit. En effet, la taille de la fenêtre de congestion définit le nombre maximal de paquets en attente d'acquittement. Pour une fenêtre de congestion W , le débit est limité à $\frac{W}{RTT}$ paquets par seconde.

Ce mécanisme repose sur l'hypothèse que si un paquet est perdu, c'est qu'il a été rejeté par un routeur à un endroit congestionné de la route empruntée par le paquet. Cette hypothèse est vraie dans les réseaux filaires, car les pertes de paquets dues à un lien défaillant y sont rares. En revanche, dans les réseaux sans fil, les pertes sont plus fréquentes, et n'indiquent pas nécessairement qu'un lien est défaillant de manière permanente : des pertes peuvent être provoquées par la présence d'un autre flux occasionnant

des collisions, ou par une modification temporaire de l'environnement, comme le passage d'un obstacle entre émetteur et récepteur.

Les interactions entre TCP et la couche MAC de 802.11 sont aujourd'hui bien connues et ont fait l'objet de nombreuses études. Altman et al. identifient quatre difficultés liées à l'utilisation de TCP dans les réseaux 802.11 [37]. Alors que les deux premiers problèmes sont propres aux réseaux ad hoc mobiles, les deux derniers sont des problèmes liés aux réseaux ad hoc statiques :

- TCP ne peut distinguer les pertes de paquets dues à la congestion des pertes dues à un lien défaillant,
- TCP souffre des reconstructions de route fréquentes,
- TCP souffre de la contention intra-flux,
- TCP n'est pas équitable.

Dans les deux premiers cas, TCP réagit de manière erronée aux pertes de paquets dues à un lien défaillant et affiche alors de mauvaises performances. Dans les deux suivants, c'est l'agressivité de TCP qui est en cause. Bien que de nombreuses solutions aient été proposées pour résoudre ces difficultés, on constate que de manière générale, TCP ne dispose pas d'informations suffisantes pour effectuer une régulation de débit équitable. Notamment, puisque TCP ne régule son débit qu'en fonction des pertes que lui-même observe, il ne permet pas de résoudre les problèmes asymétriques étudiés dans le chapitre 2 (stations cachées asymétriques, et scénario des trois paires).

Deuxième partie

Profiterole : Proportionnal Fair Rate
Control Protocol

Chapitre 4

Modèles et outils théoriques

Le chapitre qui suit présente les bases théoriques et les justifications des hypothèses faites et des modèles choisis pour représenter la contention dans les réseaux ad hoc dans le protocole *Profiterole*. Ces modèles sont l'objet des sections 4.1, 4.2, et 4.3. Ensuite, les sections 4.4 et 4.5 fournissent les outils théoriques permettant de résoudre le problème d'optimisation des débits selon les contraintes du modèle de contention en un algorithme distribué.

4.1 Modélisation des interactions dans les réseaux 802.11

Les chapitres 2 et 3 ont montré que, dans les réseaux ad hoc, la norme 802.11 pose des problèmes d'efficacité et d'équité en termes de bande passante. Une première solution à ces problèmes consiste à proposer une alternative à la couche MAC de IEEE 802.11 afin d'augmenter l'efficacité et l'équité dans le réseau. Bien que de nombreuses solutions améliorent nettement la situation, il est probable dans un futur proche qu'il faille s'accomoder de la norme actuelle, notamment car modifier la couche MAC implique souvent de construire de nouvelles cartes réseau.

Une autre approche consiste à considérer que 802.11 restera la norme des réseaux sans fil pour quelques temps, et de pallier ces faiblesses dans les couches supérieures, au dessus de 802.11. L'origine principale des inégalités de partage de bande passante est la congestion. Une manière de résoudre ces problèmes consiste donc à réguler les débits à la source afin de limiter la congestion dans le réseau. Le problème devient donc un problème d'allocation de ressources : pour chaque source d'un flux, à quel débit peut-elle émettre? Déterminer cette allocation nécessite l'utilisation d'un modèle de contention représentant les dépendances entre les liens réseaux. Ces dépendances sont de deux types :

- La *contention* : un émetteur ne peut émettre car la couche physique lui signale que le canal n'est pas libre (mécanisme de *Clear Channel Assessment*, i.e une ou plusieurs stations émettent à proximité, et la somme des signaux reçus dépasse le seuil de détection de porteuse.
- Les *collisions* : la réception d'un signal est brouillée par d'autres signaux de plus faible amplitude.

Les auteurs de [34] proposent deux modèles que l'on retrouve régulièrement dans la littérature : le *modèle physique*, et le *modèle protocolaire*.

Dans le modèle physique, un signal peut être reçu si le SINR¹⁵ au niveau du récepteur est supérieur à un certain seuil. Ainsi, on peut déterminer si deux transmissions pourront avoir lieu simultanément. Soit on calcule les SINR au niveau de leurs récepteurs respectifs en fonction des distances qui les séparent d'autres émetteurs selon un certain modèle de propagation - cette méthode suppose donc de connaître précisément les positions des stations. On peut aussi se reposer sur des mesures de puissance faites par chaque station. Ainsi, on s'affranchit du modèle de propagation. Ici, les ressources du réseau sont les liens, i.e des couples émetteur-récepteur orientés.

Dans le modèle protocolaire, une transmission se déroule correctement si aucune émission n'a lieu simultanément à une distance inférieure à une distance seuil de l'émetteur comme du récepteur. Ce dernier

¹⁵ pour rappel, le rapport signal sur bruit plus interférences

modèle, plus simple, regroupe contention et interférences en une seule interaction, et c'est la distance entre les stations qui détermine si cette interaction aura lieu. Dans ce modèle, le sens des transmissions n'a plus d'importance, puisque émetteur et récepteur jouent un rôle identique. Les ressources sont donc les liens, sans distinction d'orientation.

Enfin, pour le protocole *Profiterole*, nous avons choisi un troisième modèle, où les ressources sont cette fois-ci les stations du réseau, et où l'on ne modélise que la contention. Les détails de ce modèle seront expliqués lors du chapitre 5.

Quel que soit le modèle choisi, on va représenter les dépendances entre ressources - qu'elles soient liens ou stations - sous la forme d'un graphe, appelé *graphe de conflit*.

4.1.1 Définitions

Définition 1 (Graphe de conflit). *Dans un graphe de conflit, ou graphe de contention, les sommets du graphe sont les ressources, et il y a une arête entre deux sommets si ces deux ressources ne peuvent pas être activées simultanément.*

Définition 2 (Allocation). *Une allocation $X = \{x_1, x_2, \dots, x_n\} \in [0, 1]^n$ est un vecteur de poids associés à chaque sommet du graphe de conflit, représentant le débit de ce sommet rapporté à une capacité unitaire.*

Définition 3 (Ordonnancement). *Un ordonnancement d'une allocation X est une application qui à chaque nœud i du graphe de conflit associe un ensemble d'intervalles disjoints de $[0..1]$ de longueur totale x_i .*

Définition 4 (Allocation / Ordonnancement faisable). *Une allocation est dite faisable s'il existe un ordonnancement des émissions tel que deux sommets voisins dans le graphe de conflit ne sont pas activés simultanément, i.e leurs ensembles d'intervalles ne se recouvrent pas.*

Dans les sections suivantes, on va étudier différents types de contraintes permettant de déterminer si une allocation est faisable. Le graphe de conflit est désormais noté G .

4.1.2 Contraintes sur les ensembles indépendants maximaux

Théorème 1. *Soit I l'ensemble des ensembles indépendants maximaux du graphe G . Alors l'allocation $X = \{x_1, x_2, \dots, x_n\}$ est faisable si et seulement si il existe $z = \text{Card}(I)$ poids non négatifs $\lambda_1, \lambda_2, \dots, \lambda_z$ tels que :*

$$\sum_{i=1}^z \lambda_i \leq 1 \text{ et } \forall j \in [1, n] \quad x_j \leq \sum_{i/j \in I_i} \lambda_i \quad (4.1)$$

Ces contraintes sont appelées *contraintes sur les ensembles indépendants maximaux*.

Démonstration.

Existence d'un ordonnancement :

Soit X une allocation vérifiant les contraintes par ensembles indépendants. On prouve l'existence d'un ordonnancement de X par construction. On divise l'unité de temps en z slots de longueurs respectives λ_i . La contrainte $\sum_{i=1}^z \lambda_i \leq 1$ assure qu'une telle division existe. Soit alors l'ordonnancement où chaque nœud est activé durant les slots associés aux ensembles indépendants dont il fait partie. Par construction, deux nœuds voisins dans le graphe de conflit ne peuvent être actifs simultanément.

Optimalité :

Etant donné un ordonnancement d'une allocation X sur une durée T . A tout instant, l'ensemble des nœuds actifs est un ensemble indépendant, et il existe un ensemble indépendant maximal i qui le contient. λ_i doit être choisi tel que $T \cdot \lambda_i$ soit la durée totale d'activation de cet ensemble indépendant maximal. On vérifie par construction que $\sum_{i=1}^z \lambda_i \leq 1$.

Tout ordonnancement faisable vérifie donc les contraintes par indépendants maximaux. \square

Discussion. Le calcul des ensembles indépendants maximaux est un problème NP-complet dans un graphe quelconque. D'autre part, ces contraintes sont difficiles à utiliser dans un cadre de réseaux ad hoc, où l'on privilégie les solutions distribuées, car les ensembles indépendants sont composés de stations arbitrairement éloignées dans le réseau. C'est pourquoi on utilise plus facilement d'autres types de contraintes sur des structures plus locales telles que les contraintes par lignes et par cliques maximales.

4.1.3 Contraintes par lignes

Definition 5. Si A est la matrice d'adjacence¹⁶ du graphe G , et X une allocation sur G , alors les contraintes par lignes sont définies par :

$$AX \leq \mathbf{1} \quad (4.2)$$

On peut réécrire ces contraintes sous une forme peut être plus intuitive :

$$\forall i \sum_{j \in V(i)} x_j \leq 1 \quad (4.3)$$

où $V(i)$ est le voisinage de la ressource i dans G .

Théorème 2. Les contraintes par lignes sont suffisantes mais pas nécessaires pour garantir qu'il existe une allocation faisable.

Démonstration. Soit X une allocation vérifiant ces contraintes. Supposons $x_i \in \mathcal{Q}$, x_i de la forme $\frac{p_i}{q_i}$ tels que $p_i \in \mathcal{Z}$ et $q_i \in \mathbb{N}^*$. Soit q le plus petit commun multiple des q_i . On réécrit alors $x_i = \frac{k_i}{q}$. Alors les contraintes par ligne peuvent être réécrites :

$$\sum_{j \in V(i)} k_j \leq q \quad (4.4)$$

où $V(i)$ est le voisinage de i dans G .

On construit G' le graphe où l'on divise chaque sommet i de G en q sous-sommets tous voisins et ayant de plus les mêmes voisins que i . D'après l'équation 4.4, chaque sommet n'a pas plus de $q - 1$ voisins dans G' . On peut donc trouver un coloriage à q couleurs du graphe G' , en utilisant un algorithme de type glouton. On associe alors un slot de durée $1/q$ à chaque couleur, et on construit un ordonnancement où chaque sommet i du graphe de conflit est actif durant les slots correspondant aux k_i couleurs qui lui sont attribuées dans G' . On vérifie facilement qu'on a alors un ordonnancement faisable. \square

Discussion Les contraintes par ligne peuvent être arbitrairement surcontraintes dans un graphe de conflit quelconque. Si G est une étoile¹⁷ (voir figure 4.1), alors les contraintes par lignes imposent $\sum_{i \in G} x_i \leq 1$, soit, pour tout i , $x_i = 1/n$, qui est très loin d'être une allocation maximale. Une allocation de la forme $\{1 - \tau, \tau, \tau, \dots\}$ où $\tau \in [0, 1]$ est faisable : le nœud central émet durant $1 - \tau$, puis les autres nœuds émettent simultanément pendant une durée τ . En choisissant n arbitrairement grand, on peut rendre les contraintes par lignes aussi surcontraintes que l'on veut.

4.1.4 Contraintes par cliques

Definition 6 (Clique). Soit $G(V, E)$ un graphe non orienté. Un sous-graphe induit¹⁸ C est une clique si et seulement si C est complet¹⁹.

¹⁶ $A_{ii} = 1$ et $A_{ij} = 1$ si les ressources i et j sont adjacentes, et 0 sinon

¹⁷Une étoile est un graphe à n sommets où l'un des sommets est voisin des $n - 1$ autres sommets

¹⁸Soit un sous-ensemble de sommets $V' \in V$, le sous-graphe induit par V' est le graphe $G'(V', E')$ où deux sommets de V' sont reliés dans G' si et seulement s'ils sont reliés dans G . $\forall e = (v_1, v_2) \in V'^2 \Rightarrow (e \in E \Leftrightarrow e \subset E')$

¹⁹Un graphe est complet si tous ses sommets sont reliés deux à deux

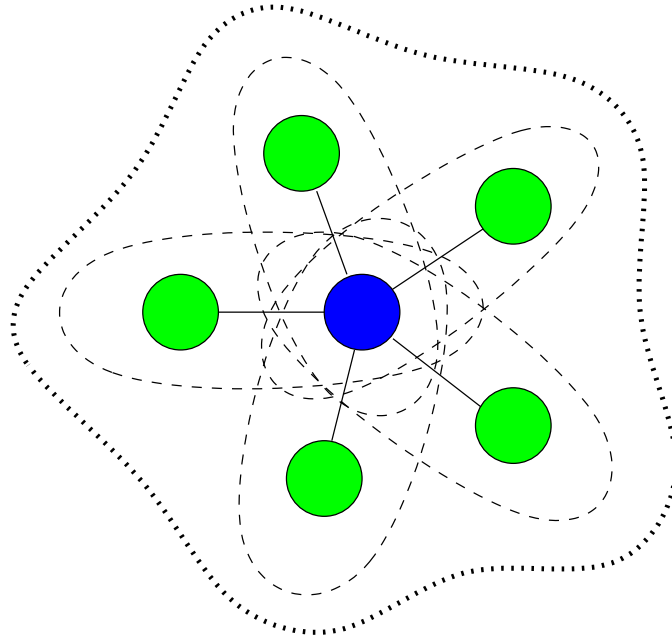


FIG. 4.1 – Graphe de conflit en étoile : les contraintes par lignes sont en pointillés courts (ceux entourant l’ensemble de la figure), et les contraintes par cliques sont en pointillés longs. Toute allocation Pareto optimale est de la forme $\{1 - \tau, \tau, \tau, \dots\}$ où $\tau \in [0, 1]$.

Le problème de la clique maximum - trouver dans un graphe la clique comprenant le plus de sommets - est un problème NP-complet dans le cas général. Les cliques ne semblent donc pas plus simples que les ensembles indépendants. Cependant, comme on va le voir dans la section sur les Unit Disk Graphs, le problème est simplifié dans les graphes de conflit des réseaux ad hoc, puisqu’une clique correspond à un ensemble de liens ou de stations proches les uns des autres.

Definition 7 (Contraintes par cliques). *Soit K l’ensemble des cliques maximales de G , et X une allocation sur G . Alors les contraintes par cliques sont définies par :*

$$\forall k \in K \quad \sum_{s_i \in k} x_i < 1$$

Théorème 3 (Nécessité des contraintes par cliques). *Les contraintes par cliques sont une condition nécessaire pour qu’un ordonnancement de X existe.*

En effet, dans une clique, pas plus d’une ressource ne peut être activée à la fois. Il est donc naturel que la somme des émissions dans une clique n’excède pas la capacité. Dans le cas du graphe de conflit en étoile, évoqué dans la section précédente, les cliques sont les paires de sommets comprenant le sommet central. Les contraintes par cliques sont dans ce cas nécessaires *et suffisantes*. En revanche, dans le cas de la figure 4.2, où les cliques maximales sont représentées en pointillés, l’allocation où pour tout i , $x_i = 1/2$, qui vérifie les contraintes, n’est pas ordonnançable. Ce graphe de conflit est appelé *trou de longueur cinq* ; il lui manque la propriété de *perfection*, décrite plus bas.

4.1.5 Graphes parfaits

Definition 8 (Graphe parfait). *Un graphe est dit parfait si et seulement si son nombre chromatique est égal à la taille de sa plus grande clique.*

On rappelle que le nombre chromatique est le nombre minimal de couleurs nécessaire pour colorier chaque sommet d’un graphe de telle sorte que deux sommets voisins soient de couleur différente. Le problème du coloriage d’un graphe est intimement lié aux problèmes d’ordonnancement dans les graphes de

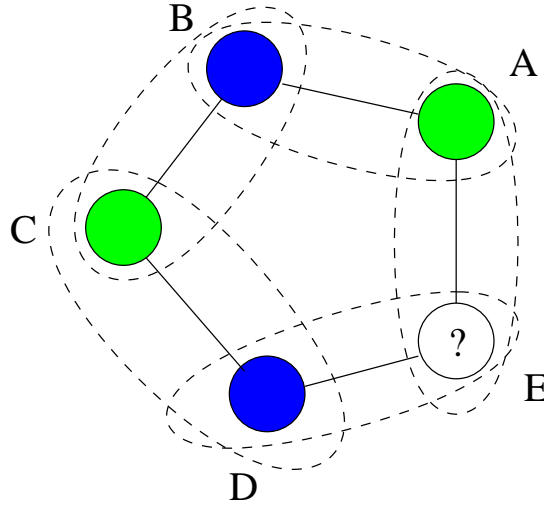


FIG. 4.2 – L’allocation $X = \{\forall i, x_i = 1/2\}$ n’est pas ordonnable : si le nœud B est actif pendant la moitié du temps (couleur bleue), alors les nœuds A et C doivent être actifs pendant le reste du temps (couleur verte). On arrive à une impossibilité pour le nœud E .

conflit : les différentes couleurs correspondent à des intervalles de temps dans le problème d’ordonnancement, et l’on veut interdire que deux sommets voisins dans le graphe de conflit aient la même couleur, et donc qu’ils soient actifs simultanément. Mais à quoi ressemble un graphe parfait ? Le théorème suivant, qui est resté une conjecture de Claude Berge, l’inventeur des graphes parfaits, n’a été démontré que très récemment dans une démonstration d’une centaine de pages. Ce théorème très puissant nous donne une idée assez précise de la structure des graphes parfaits :

Théorème 4 (Théorème fort des graphes parfaits). *Un graphe est parfait si et seulement si ni lui, ni son complémentaire ne contiennent de cycle sans corde induit de longueur impaire et supérieure à cinq.*

Je rappelle qu’un sous-graphe induit d’un graph $G(V, E)$ est un graphe G' composé d’un sous-ensemble $V' \in V$ de sommets de G et de toutes les arêtes entre les sommets de V' qui apparaissent dans le graphe G . Un cycle sans corde est aussi appelé *trou*. Ces trous sont facilement identifiables dans les graphes de conflit des réseaux ad hoc.

Théorème 5 (Condition de suffisance des contraintes par cliques [35]). *Les contraintes par cliques sont nécessaires et suffisantes si et seulement si G est un graphe parfait.*

Démonstration. On va utiliser une méthode semblable à celle utilisée pour la suffisance des contraintes par lignes :

1. Supposons qu’on ait $\forall i, x_i \in \mathbb{Q}$ et posons $x_i = \frac{p_i}{q}$. On construit le graphe G' où le sommet i est remplacé par la clique K_{p_i} (une clique de taille p_i). D’autre part, on ajoute des arêtes entre tous les sommets de deux de ces cliques voisines dans G . Le graphe G' reste parfait par cette transformation (ce qui se démontre, par exemple, par le théorème des graphes parfaits : aucun cycle sans corde de longueur impaire).

2. Les contraintes deviennent alors :

$$\forall k \in K \sum_{s_i \in k} p_i < q$$

3. La taille de la clique maximum dans G' est donc majorée par q . G' étant parfait, il existe un q -coloriage de G' .
4. On divise alors le temps en q slots, et la station s_i peut émettre pendant les slots correspondant à ses couleurs.

On peut ainsi construire un ordonnancement des émissions pour tout X vérifiant les contraintes par cliques. Ces contraintes sont donc suffisantes en plus d'être nécessaires dans le cas des graphes parfaits. \square

Bien que cette méthode de construction prouve l'existence d'un ordonnancement, il peut être difficile à trouver en pratique. Pour une valeur de $X \in \mathbb{R}^n$, on peut trouver une valeur $X' \in \mathbb{Q}^n$ aussi proche que l'on veut de X (\mathbb{Q} est dense dans \mathbb{R}), mais plus l'approximation est précise, et plus la valeur de q est élevée. Et en pratique, le temps n'est pas divisible à l'infini. Cependant, on va voir dans la suite que les graphes de conflit des réseaux ad hoc sont *presque* parfaits. A défaut de nous donner un ordonnancement faisable, ce théorème nous donne donc une idée de la faisabilité des allocations vérifiant les contraintes par cliques.

Pour résumer, nous avons vu trois types de contraintes :

- les contraintes par ensembles indépendants,
- les contraintes par lignes,
- les contraintes par cliques,

De ces trois types de contraintes, seul le premier donne un ensemble de conditions nécessaires *et* suffisantes. Malheureusement, les contraintes par ensembles indépendants sont a priori difficiles à utiliser dans un environnement distribué. Sur des graphes de conflit quelconques, les deux autres types de contraintes peuvent être pour les unes, arbitrairement surcontraintes, et pour les autres, arbitrairement sous-contraintes. Cependant, les graphes de conflit des réseaux ad hoc ne sont pas quelconques. Si l'on admet quelques hypothèses sur la propagation radio, on constate qu'ils ont des propriétés qui rendent ces contraintes nettement plus précises.

4.2 Unit Disk Graphs

4.2.1 Introduction

Les Unit Disk Graphs (UDG) ont été introduits dans les années 80-90 [22] pour modéliser les interactions dans les réseaux sans fil. Les sommets d'un tel graphe peuvent être les stations du réseau, ou ses liens, selon ce que l'on souhaite modéliser. Deux sommets sont reliés dans ce graphe si ces stations sont à une distance inférieure à une distance seuil. Cette distance peut être la distance de communication, de détection de porteuse, ou d'interférence. Pour simplifier la modélisation, on suppose que la portée de communication - respectivement la capacité à provoquer des interférences - est la même dans toutes les directions et pour toutes les stations. De nombreux problèmes classiques des graphes ont des applications utiles dans les réseaux sans fil :

- Coloriage de graphe : pour éviter les interférences, on cherche une attribution des canaux de sorte à ce que deux stations voisines utilisent des canaux différents.
- Minimum dominating set : permet de trouver un ensemble de stations qui joueront le rôle d'arbitres locaux, de sorte que chaque station du réseau soit à portée de communication d'au moins un arbitre.
- Connected dominating set : il s'agit du même problème, avec la contrainte supplémentaire que l'ensemble des arbitres doit former un sous-graphe connexe.
- All Maximal cliques : utilisé pour les équations de contraintes de partage de la bande passante (pas plus d'une station dans une clique ne peut émettre au même instant).
- All Maximal independent set : une autre approche pour générer les contraintes. Cette dernière est plus difficile à utiliser car les ensembles indépendants induisent des algorithmes centralisés par opposition aux cliques, qui sont des structures locales, et induisent des algorithmes distribués.

4.2.2 Définitions et propriétés

Definition 9 (Unit Disk Graph). *Etant donné P un ensemble de points dans le plan, on construit son Unit Disk Graph associé $G(V, E)$ où*

- à chaque point de P est associé un sommet dans V ,
- il existe une arête entre 2 sommets si la distance Euclidienne entre les deux points associés est au plus 1.

Un UDG peut aussi être vu comme le graphe d'intersection d'un ensemble de disques de rayon $\frac{1}{2}$. On peut aussi généraliser les Unit Disk Graphs à plus de deux dimensions, en remplaçant les disques par des boules. Dans la suite, on ne s'intéresse qu'au cas à deux dimensions.

Propriété 1. *Soit $G(V, E)$ un UDG. $\forall x \in V$, on note $N(x)$ l'ensemble des voisins de x , x compris. Alors $N(x)$ ne contient pas d'ensemble indépendant de taille supérieure à 5.*

La preuve est géométrique et immédiate : si x a six voisins ou plus, alors deux d'entre eux sont voisins. On peut donc en déduire une sous-structure interdite des UDG :

Propriété 2. *Soit $G(V, E)$ un UDG. Alors G ne contient pas de sous-graphe isomorphe à $K_{1,6}$ ²⁰.*

Propriété 3. *Les UDG ne sont pas des graphes parfaits.*

Le trou c'est-à-dire le cycle sans corde, de longueur cinq est un UDG, mais c'est aussi une sous-structure interdite des graphes parfaits. On peut en effet placer cinq points aux sommets d'un pentagone régulier de sorte que le côté du pentagone soit moins de un, mais que deux sommets non voisins soient éloignés de plus de un.

Propriété 4. *Les UDG ne sont pas des graphes planaires.*

La clique de taille cinq, K_5 est un UDG. Hors, c'est une sous-structure interdite des graphes planaires. En revanche, un graphe planaire de degré inférieur à quatre peut être plongé dans un UDG (théorème de Valiant dans [64], propriété qui est utilisée pour démontrer la NP-complétude des problèmes cités ci-dessous.

Quelques résultats de complexité

La majorité des problèmes NP-durs dans les graphes quelconques le restent dans les UDG. Colbourn et Clarke [22] donnent en 1990 la complexité de quelques problèmes dans les UDG, consignées dans le tableau 4.1. En pratique, on dispose de bonnes heuristiques, notamment pour les problèmes de clique maximale, et d'ensemble dominant dans [51].

Nombre chromatique	NP-complet
Clique maximale	<i>Polynomial</i>
Ens. indépendant	NP-complet
Ensemble dominant	NP-complet
Ensemble dominant connecté	NP-complet
Circuit Hamiltonien	NP-complet
Reconnaissance	NP-dur [12]

TAB. 4.1 – Quelques problèmes des UDG et leur complexité

4.2.3 Borne sur le degré d'imperfection

Soit $G(V, E)$ un graphe, et X un vecteur entier de dimension $n = \text{Card}(V)$. On cherche à savoir combien de couleurs différentes sont nécessaires pour colorier le graphe G de sorte à ce que chaque sommet i soit colorié avec x_i couleurs différentes, et que deux voisins dans G n'ont aucune couleur en commun. Pour résoudre ce problème, on construit naturellement le graphe G_X , graphe où l'on remplace chaque sommet i de G par la clique de taille x_i . Le problème énoncé précédemment revient donc à trouver le nombre chromatique de G_X .

²⁰ $K_{n,m}$ est la notation usuelle pour le graphe bipartite composé de $n + m$ sommets et $n * m$ arêtes où chacun des n sommets est relié aux m autres sommets. $K_{1,6}$ est donc un graphe en étoile avec un sommet au centre, et six branches.

Definition 10. Le degré d'imperfection d'un graphe G est défini dans [32] par la formule suivante :

$$\text{imp}(G) = \lim_{K \rightarrow \infty} \max \left\{ \frac{\chi(G_X)}{\omega(G_X)}, \quad X \in \mathbb{N}, \max_{i \leq n} (x_i) = K \right\}$$

Intuitivement, il s'agit de trouver le rapport entre taille de la clique maximum et nombre chromatique de G_X dans le pire cas. Ce rapport indique dans quelle proportion les contraintes par cliques peuvent être sous-contraintes.

Remarques :

- $\text{imp}(G) = 1$ si et seulement si G est parfait
- $\text{imp}(G) > 1$ sinon.
- Le degré d'imperfection dans un graphe quelconque peut être arbitrairement grand (voir par exemple la construction de Mycielski dans Introduction to Graph Theory de Douglas West [69], page 205-206).

Théorème 6. Si G est un UDG, alors :

$$\text{imp}(G) \leq 1 + \frac{2}{\sqrt{3}}$$

Ce théorème est prouvé dans [35]. Les auteurs suggèrent que cette borne soit large, et que la borne théorique est $\frac{3}{2}$, borne atteinte par le trou de taille 5, où $\chi(G) = 3$ et $\omega(G) = 2$. Prouver que la borne est de $\frac{3}{2}$ est un problème ouvert.

Ce résultat permet d'envisager un nouveau type de contraintes qui, à la différence des contraintes par cliques, sont, elles, suffisantes.

Théorème 7 (Scaled Clique Constraints [35]). Soit G un UDG. Soit K l'ensemble des cliques maximales de G , et C la capacité du medium radio. Alors les contraintes suivantes sont suffisantes :

$$\forall k \in K \quad \sum_{s_i \in k} x_i < \frac{C}{1 + \frac{2}{\sqrt{3}}}$$

4.2.4 Conclusion

Bien que les graphes de conflit des réseaux ad hoc ne soient pas des UDG, puisque il n'est pas réaliste de considérer que la zone de communication est un disque parfait, ces résultats donnent un peu plus de crédit aux contraintes par cliques. D'autre part, même si la borne théorique sur le degré d'imperfection des UDG est au mieux $\frac{3}{2}$, les auteurs de [50], en calculant le degré d'imperfection sur des UDG obtenus en plaçant aléatoirement des points sur le plan, ont montré que ce degré est proche de 1 en moyenne. Intuitivement, on comprend qu'il est assez improbable d'observer un trou de longueur 5 (et son complémentaire l'est encore plus) dans un Unit Disk Graph, et d'autant moins probable que le réseau est dense.

4.3 Voisinage à k sauts

La distance entre deux stations est un indicateur assez peu fiable pour déterminer si celles-ci sont à portée de communication, à cause de l'imprévisibilité du medium radio, et surtout, c'est une information difficile à obtenir. Afin de s'affranchir de cette difficulté, de nombreux protocoles utilisent la distance en nombre de sauts radio, i.e. la longueur de la route la plus courte menant d'une station à une autre.

Les expériences qui suivent décrivent la relation entre distance Euclidienne et nombre de sauts en fonction de la densité de stations dans le réseau. En effet, plus le réseau est dense, et plus grande est la probabilité qu'une station située à k sauts soit à une distance avoisinant k fois la distance de communication.

n	100	200	300	500	800	2000
δ	π	2π	3π	5π	8π	20π

FIG. 4.3 – Correspondance entre n (nombre de nœuds) et la densité δ (nombre moyen de nœuds par zone de communication, i.e. sur une surface πd^2)

Distance	1 saut	2 sauts	3 sauts	4 sauts	5 sauts	6 sauts	7 sauts	Total
0-25	92							92
25-50	304							304
50-75	460							460
75-100	638							638
100-125		895	6	1				902
125-150		1095	54	1	2			1152
150-175		976	311	8	1	1		1297
175-200		354	1074	21	1			1450
200-225			1505	140	7	2		1654
225-250			1387	499	10	1		1897
250-275			700	1279	35	4		2018
275-300			79	1939	158	8	1	2185
300-325				1852	610	14	2	2478
325-350				1173	1517	49	5	2744
350-375				311	2387	184	5	2887
375-400				10	2415	630	20	3076

FIG. 4.4 – Répartition en nombre de sauts en fonction de la distance au centre du carré. Résultats pour $n = 500$. Par exemple, sur l'ensemble des 100 expériences, on a compté, dans la fenêtre 125-150 m, 1095 stations à deux sauts de distance, 54 à 3 sauts, 1 à 4 sauts, et 2 à 5 sauts sur l'ensemble des 100 expériences.

Les résultats de la figure 4.5 ont été obtenus en plaçant n stations aléatoirement dans un carré de taille 1000×1000 selon un processus aléatoire uniforme. En supposant une distance de communication $d = 100m$, on a calculé pour chaque station la route la plus courte en nombre de sauts au centre du carré, qu'on met en rapport avec la distance réelle. Pour chaque valeur de densité δ (voir correspondance entre densité et nombre de stations dans le tableau 4.3), l'expérience a été renouvelée cent fois. Pour chaque tranche de 25 m en distance Euclidienne au centre, on observe la distribution des stations en nombre de sauts. Le tableau 4.4 donne les résultats agrégés de l'expérience avec $n = 500$. La figure 4.5 donne ces résultats sous forme graphique.

On peut tirer deux conclusions de ces expériences :

- Pour une densité inférieure à 2π (soit moins de 6 nœuds à portée de communication en moyenne), le graphe de communication n'est pas connexe, et de nombreux nœuds sont inaccessibles (représentés dans la catégorie *zero saut*). Pour une telle densité, la métrique du nombre de sauts ne peut être utilisée.
- Pour des densités plus fortes, le nombre de sauts est un indicateur assez fiable de la distance entre deux stations, mais devient de moins en moins fiable à mesure qu'on augmente le nombre de sauts. Par exemple, pour $\delta = 2\pi$, 10% des nœuds situés à une distance entre 0 et 200 m du centre du carré, soit entre d et $2d$, se trouvent à plus de 5 sauts. Pour $\delta = 5\pi$, 80% des nœuds à une distance comprise entre 0 et 200m sont à une distance de moins de 2 sauts, et la grande majorité des nœuds à 4 sauts ou moins sont situés à moins de 300 m.

4.4 Equité et Fonctions objectif

Les sections précédentes ont montré diverses manières de représenter les interactions dans les réseaux ad hoc sous formes de contraintes. Ces contraintes définissent un domaine faisable, c'est-à-dire l'ensemble des valeurs possibles X des vecteurs de débits qui vérifient les contraintes. Dans notre cas, on cherche les valeurs maximales de X , celles pour lesquelles les contraintes sont saturées, et donc aux limites du domaine faisable. Ces valeurs-limites sont dites *Pareto-optimales*. L'allocation qui maximise le débit total, l'allocation max-min, ou l'allocation proportionnelle-équitable, sont autant d'allocations Pareto-optimales. Le rôle de la section suivante est de formaliser ces concepts.

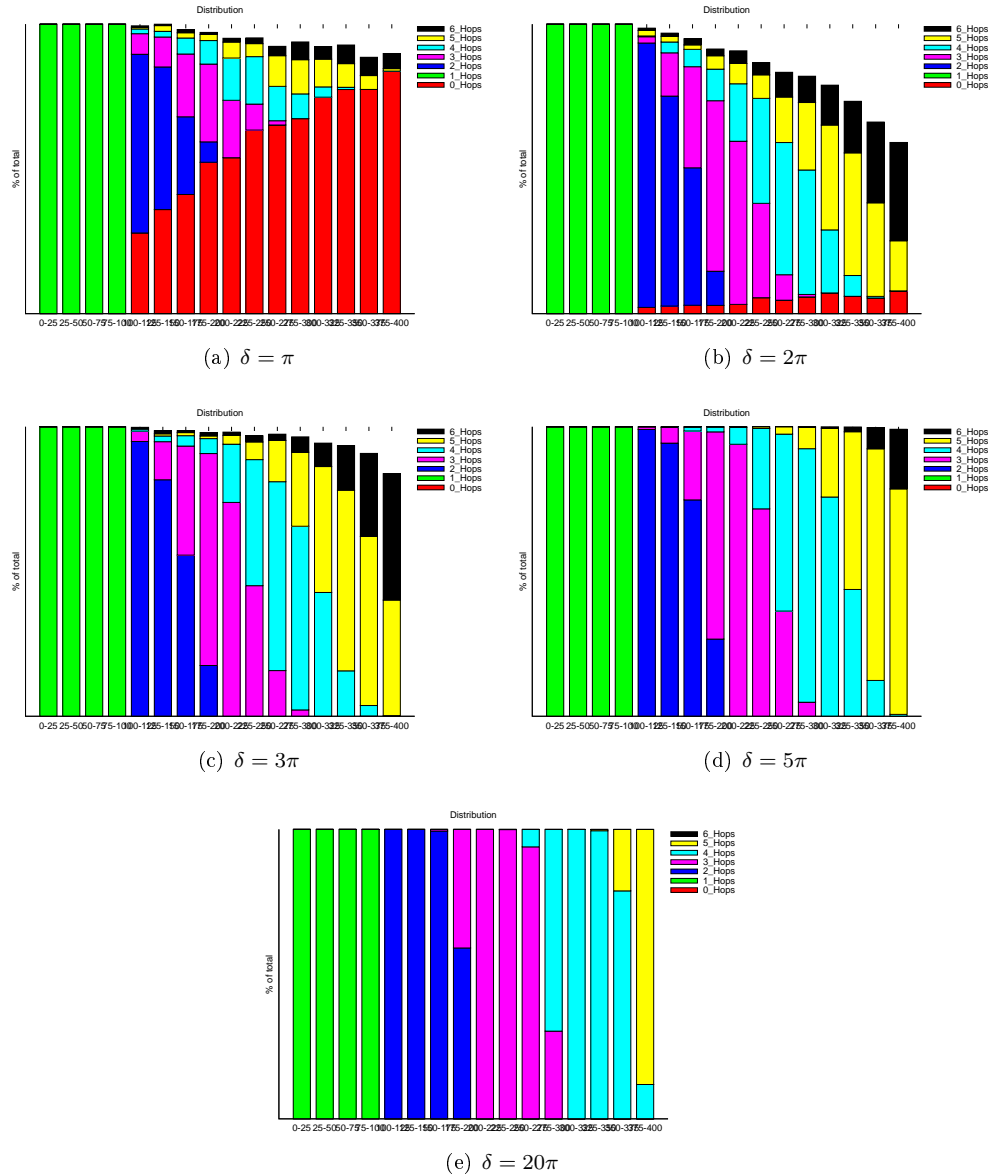


FIG. 4.5 – Distribution des nœuds en nombre de sauts en fonction de la distance pour plusieurs densités. Les nœuds non connectés sont dans la catégorie *0 hops*. Les nœuds à plus de 6 hops de distance ne sont pas représentés. Chaque barre de l’histogramme correspond à une tranche de 25 m en distance Euclidienne. Les différentes couleurs permettent d’observer la proportion de stations à un saut, à deux sauts, etc... parmi ces stations. Par exemple, pour la densité la plus basse, $\delta = \pi$, une station à une distance entre 100 m et 125 m a une probabilité d’environ 60% d’être atteignable en deux sauts, environ 10% d’être à trois sauts ou plus, et 30% pour qu’elle ne soit pas joignable du tout (représentés en rouge sur le graphique).

4.4.1 Définitions et notations

Définition 11 (Domaine faisable). Soit l'espace vectoriel $E = \mathbb{R}^n$ de dimension $n \in \mathbb{N}^*$. Un élément de E est noté $X = (X_1, X_2, \dots, X_n)$. On munit E de la relation d'ordre partiel \preceq telle que

$$X \preceq Y \Leftrightarrow \forall i \in [1..n] \quad X_i \leq Y_i$$

Soit un entier $m \in \mathbb{N}^*$ et une fonction C de E dans \mathbb{R}^m qu'on appellera fonction de contrainte. Alors le domaine faisable est défini par :

$$\{X \in E \text{ tel que } C(X) \preceq 0_m\}$$

Définition 12 (Pareto optimalité). $X_p \in E$ est dit Pareto-optimal selon la fonction contrainte C si et seulement si :

$$\forall Y \in E \Rightarrow (X_p \preceq Y \Rightarrow C(Y) \not\preceq 0_m)$$

La notion de Pareto-optimalité provient du domaine de l'économie, où un optimum de Pareto est un état dans lequel on ne peut pas améliorer le bien-être d'un individu sans détériorer celui d'un autre. Ici, les individus sont les n composantes du vecteur X_p , et il n'est pas possible d'augmenter une composante sans qu'une contrainte ne soit dépassée. En revanche, cela n'implique pas que $C(X_p) = 0$.

Théorème 8. Si C est croissante (i.e. toutes ses composantes sont croissantes), alors le domaine faisable est un ensemble convexe.

Démonstration. Soit D le domaine faisable pour la fonction contrainte C , dont chacune des composantes $C_i, i \in 1..n$ est croissante. Soit $X, Y \in D^2$ deux points du domaine faisable. On va montrer que :

$$\forall \alpha \in \mathbb{R}, 0 < \alpha < 1 \Rightarrow C(Z) \preceq 0_m \quad \text{avec} \quad Z = \alpha X + (1 - \alpha)Y$$

Pour tout $i \in 1..n$, on compare les $i^{\text{èmes}}$ composantes x_i et y_i des vecteurs X et Y . Sans perte de généralité, supposons que $x_i < y_i$. Alors $x_i < z_i < y_i$, et par croissance de C_i , on a bien $C_i(Z) \leq C_i(Y) \leq 0$.

On a donc $C(Z) \preceq 0_m$, et Z appartient au domaine faisable. \square

Fonction objectif Les contraintes définissent donc un ensemble de points atteignables (i.e. qui vérifient les contraintes), et il s'agit de trouver dans cet ensemble le ou les points qui maximisent - ou minimisent - une fonction appelée fonction objectif. Dans le contexte de l'optimisation de la répartition de bande passante entre flux, la fonction objectif est une fonction prenant une allocation pour argument, et renvoyant un nombre réel qui doit refléter la qualité d'une allocation de bande passante, et la satisfaction des utilisateurs du réseau. Je présente ci-après quelques fonctions objectifs utilisées couramment. Pour illustrer leurs caractéristiques, nous allons les appliquer au scénario des trois paires, présenté dans la section 2.3.2. Soient x_a, x_b , et x_c les proportions de temps pendant lesquelles les paires A, B , et C peuvent transmettre. A et C sont les paires extérieures, et B la paire centrale. On rappelle que selon le mécanisme CSMA, B ne peut émettre quand A ou C émettent. Ainsi, les contraintes sur x_a, x_b et x_c s'écrivent :

$$\begin{cases} x_a + x_b \leq 1 \\ x_b + x_c \leq 1 \end{cases}$$

4.4.2 Maximisation de l'utilisation du réseau

Ce critère va généralement dans le sens de l'exploitant du réseau, ou du fournisseur d'accès, qui cherche à ce que son réseau soit utilisé au maximum. La fonction objectif utilisée est donc :

$$U_{\text{util}}(x) = \sum_{i \in [1..n]} x_i$$

Dans le cas des trois paires, la solution qui maximise $x_a + x_b + x_c$ est $(1, 0, 1)$. La paire centrale n'a donc pas de bande passante avec ce critère, ce qui est à peu de choses près l'allocation qu'on obtient avec 802.11. De manière générale, ce critère est le moins équitable qui soit puisque pour maximiser l'utilisation du réseau, il faut d'abord servir les utilisateurs qui consomment le moins de ressources. Ici, la paire B consomme les ressources des paires A et C , et est donc servie en dernier.

4.4.3 Équité max-min

Une allocation X est dite *max-min équitable* si elle vérifie la propriété :

$$\forall Y \in D \Rightarrow y_k > x_k \Rightarrow \exists s \in [1..n] \text{ tel que } x_s \geq x_k \text{ et } y_s < x_s$$

Intuitivement, l'équité max-min, au contraire de la précédente, est *la plus équitable possible*. C'est l'équité pour laquelle on ne peut augmenter le gain d'un utilisateur sans réduire celui d'un utilisateur qui était déjà défavorisé par rapport au premier. L'allocation max-min équitable peut être obtenue avec l'algorithme glouton de type *water-filling* : on augmente progressivement la bande passante allouée à chacun des utilisateurs jusqu'à ce qu'une contrainte soit saturée, puis on recommence l'opération avec les utilisateurs n'apparaissant pas dans ces contraintes, jusqu'à ce que tous les utilisateurs apparaissent dans au moins une contrainte saturée.

Selon la définition précédente, l'allocation max-min équitable est unique. Il est cependant courant de parler d'allocation max-min équitable pour une allocation Pareto-optimale qui maximise la fonction objectif suivante :

$$U_{\max\text{-min}}(x) = \min_{i \in [1..n]} x_i \quad (4.5)$$

Dans le scénario des trois paires, l'allocation max-min équitable est $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$: la paire B émet pendant la moitié du temps, et les paires A et C émettent simultanément pendant l'autre moitié.

4.4.4 Équité proportionnelle

L'équité proportionnelle, introduite par Kelly, est un compromis entre les deux schémas ci-dessus. La fonction objectif utilisée est :

$$U_{\text{prop}}(x) = \sum_{i \in [1..n]} \ln x_i$$

Appliquée au scénario des trois paires, l'allocation proportionnelle-équitable est $(\frac{2}{3}, \frac{1}{3}, \frac{2}{3})$.

4.4.5 Synthèse des types d'allocation

Le tableau 4.2 résume quelques caractéristiques des allocations obtenues dans le scénario des trois paires. Ce tableau montre qu'il existe un compromis entre équité, représenté par l'indice de Jain, et efficacité, mesurée avec la somme des débits du réseau. Mo et Walrand ont proposé une famille de fonctions objectif qui contiennent les trois critères précédents et illustrent bien ce compromis [53], résumé dans la figure 4.6 :

$$U_\alpha(x) = \begin{cases} \frac{1}{1-\alpha} \sum x_i^{1-\alpha} & \text{si } \alpha \in [0, +\infty[\setminus 1 \\ \sum \ln x_i & \text{sinon} \end{cases} \quad (4.6)$$

A quelques constantes près, qui ne changent cependant pas l'ordre dans lequel les allocations sont classées, on retrouve les différents fonctions objectifs citées plus haut :

- $\alpha = 0$ correspond à la maximisation du débit total,
- quand α tend vers $+\infty$, $\min(x_i)$ devient prépondérant par rapport aux autres x_i et U_α tend vers $U_{\max\text{-min}}$ ²¹
- quand α tend vers 1, $x_i^{1-\alpha}$ est équivalent à $1 + (1-\alpha) \ln x_i$. Ainsi, aux constantes près, U_{1+} est la fonction objectif de l'équité proportionnelle,
- enfin, la fonction U_2 est appelée minimisation du délai.

²¹en multipliant par un facteur $1-\alpha$ et en l'élevant à la puissance $1/(1-\alpha)$

Type d'allocation	Max-min	Proportionnelle	Max. utilisation du réseau
Allocation	$(\frac{1}{2}, \frac{1}{3}, \frac{1}{2})$	$(\frac{2}{3}, \frac{1}{3}, \frac{2}{3})$	$(1, 0, 1)$
Utilisation du réseau $(\sum x_i)$	$\frac{2}{2}$	$\frac{5}{3}$	2
Index de Jain $(\frac{(\sum x_i)^2}{n \sum x_i^2})$	1	$\frac{25}{27}$	$\frac{2}{3}$

TAB. 4.2 – Allocations obtenues dans le scenario des trois paires avec plusieurs critères d'équité. L'index de Jain [58] est une mesure d'équité qui va de $\frac{1}{n}$ pour une allocation où toutes les ressources sont données à un seul utilisateur, à 1 pour une allocation parfaitement égale.

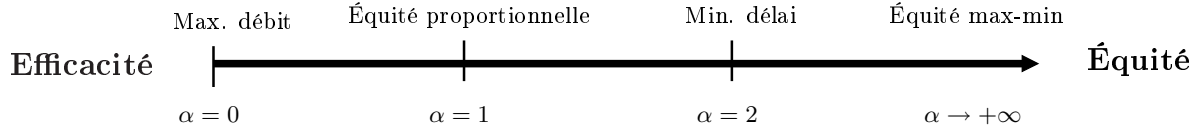


FIG. 4.6 – Compromis entre efficacité et équité dans la famille de fonctions objectifs de l'équation 4.6

4.5 Problèmes d'optimisation et relaxation Lagrangienne

Cette section présente la relaxation Lagrangienne, un outil permettant de transformer un problème d'optimisation convexe contraint en un problème dual souvent plus simple à résoudre. L'algorithme de résolution du problème d'allocation des débits de *Profiterole* repose sur cet outil. Les théorèmes présents dans cette section proviennent pour la plupart du livre sur l'optimisation convexe de Dimitri Bertsekas [9], certes brillant, mais si précis qu'il peut en être fastidieux. Comme leur énoncé, destiné à la résolution des problèmes d'optimisation dans le cas général, était parfois rébarbatif, je n'en ai conservé que le strict minimum pour l'utilisation que nous allons en faire.

4.5.1 Problèmes d'optimisation

Un problème d'optimisation s'écrit typiquement sous la forme (avec s.c pour "sous contraintes") :

$$\mathcal{P} \quad \left| \begin{array}{l} \max_{x_1, x_2, \dots, x_n} f(x_1, x_2, \dots, x_n) \\ \text{s.c} \quad \text{contrainte}(i) \quad \forall i \in [1..m] \end{array} \right.$$

Ici, il s'agit d'un problème de maximisation sous contraintes. La fonction f est appelée *fonction objectif*, et on cherche les valeurs (x_1, x_2, \dots, x_n) pour lesquelles cette fonction est maximale (ou minimale) tout en vérifiant les contraintes.

Les contraintes peuvent prendre plusieurs formes :

- contraintes en équations : $g_i(x) = 0 \quad \forall i \in [1..m]$
- contraintes en inéquations : $g_i(x) \geq 0 \quad \forall i \in [1..m]$

Definition 13 (Optimum). *La valeur x^* qui résout le programme \mathcal{P} est un optimum de la fonction f sous les contraintes du programme.*

Definition 14 (Optimum local). *Une valeur x^* est un maximum local (resp. minimum local) d'une fonction f définie sur l'ensemble convexe $S \Leftrightarrow \exists \epsilon > 0$ tel que $\forall x \in S / |x - x^*| \leq \epsilon \Rightarrow f(x) \leq f(x^*)$ (resp. $f(x) \geq f(x^*)$).*

Definition 15 (Optimum global). *Une valeur x^* est un maximum global (resp. minimum global) d'une fonction f définie sur l'ensemble convexe $S \Leftrightarrow \exists \epsilon > 0$ tel que $\forall x \in S \leq \epsilon \Rightarrow f(x) \leq f(x^*)$ (resp. $f(x) \geq f(x^*)$).*

4.5.2 Relaxation Lagrangienne

La relaxation Lagrangienne est une méthode couramment utilisée pour la résolution de problèmes d'optimisation avec contraintes. Une des premières apparitions de cette méthode dans la littérature infor-

matique est un article de Kelly [41] qui a inspiré de nombreux algorithmes d'optimisation. Je vais tâcher d'en expliquer le principe.

Pour trouver le maximum d'une fonction continue dérivable, la méthode que nous avons apprise au lycée consiste à chercher les points où la dérivée s'annule, puis à y distinguer les maxima (dérivée seconde négative) et les minima (dérivée seconde positive). Dans le cas de l'optimisation avec contraintes, cette méthode simple ne peut être utilisée. La relaxation Lagrangienne consiste à s'affranchir des contraintes et à *relâcher* le problème en introduisant les contraintes dans la fonction à maximiser.

Commençons par un problème simple, celui de la maximisation à une seule contrainte en équation :

$$\mathcal{P} \quad \left| \begin{array}{l} \max_x f(x) \\ \text{s.c. } g(x) = 0 \end{array} \right.$$

Dans la suite, on supposera que f et g sont toutes deux continûment différentiables deux fois. On suppose aussi que \mathcal{P} admet au moins une solution.

Definition 16 (Lagrangien). *Le Lagrangien \mathcal{L} est défini par :*

$$\mathcal{L}(x, \lambda) = f(x) - \lambda \cdot g(x)$$

où la variable λ est appelé multiplicateur de Lagrange²² associé à la contrainte g .

Conditions de qualification de la contrainte. Pour pouvoir utiliser la relaxation Lagrangienne dans la résolution de \mathcal{P} , il suffit que l'une des conditions suivantes soit vérifiée :

- (a) Les dérivées partielles de la fonction contrainte g évaluées à l'optimum x^* ne sont pas simultanément nulles, i.e. $\frac{\partial g}{\partial x_i}(x^*) \neq 0$ pour au moins un x_i .
- (b) La fonction contrainte g est linéaire.

Conditions du premier ordre : ²³ On suppose que la condition de qualification est vérifiée. Si le vecteur x^* est une solution du programme de maximisation \mathcal{P} , alors il existe un unique λ^* tel que x^* vérifie les $n + 1$ conditions suivantes :

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{L}}{\partial x_i}(x^*, \lambda^*) = 0 \quad \forall i \in [1..n] \\ \frac{\partial \mathcal{L}}{\partial \lambda}(x^*, \lambda^*) = 0 \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} \frac{\partial f}{\partial x_i}(x^*) - \lambda^* \frac{\partial g}{\partial x_i}(x^*) = 0 \quad \forall i \in [1..n] \\ g(x^*) = 0 \end{array} \right.$$

Conditions du second ordre pour un optimum global : Supposons qu'il existe un x^* qui vérifie les CPO. Alors :

- Si le Lagrangien \mathcal{L} est une fonction concave (ce qui est le cas en particulier si f est concave et si λ^*g est convexe), alors x^* est un maximum global.
- Si le Lagrangien \mathcal{L} est une fonction convexe alors x^* est un minimum global.

Si f est concave et si λ^*g est convexe, alors le programme d'optimisation \mathcal{P} est équivalent au programme dual \mathcal{D} suivant :

$$\mathcal{D} \quad \left| \begin{array}{l} \min_{\lambda} \max_x \mathcal{L}(x, \lambda) \end{array} \right.$$

4.5.3 Généralisation à des contraintes multiples et par inéquations

On considère le programme de maximisation à m contraintes par inéquations suivant :

$$\mathcal{P} \quad \left| \begin{array}{l} \max_x f(x) \\ \text{s.c. } g_i(x) \leq 0 \quad \forall i \in [1..m] \end{array} \right.$$

²²On trouve aussi le terme multiplicateur de Lagrange pour la valeur λ^* citée juste au-dessous.

²³Les conditions du premier ordre sont des conditions *nécessaires* que doit vérifier un optimum x^* , s'il existe. Les conditions du second ordre garantissent que les conditions du premier ordre sont *suffisantes* pour que x^* soit bien solution du programme d'optimisation.

Soit x^* une solution de ce programme. Alors deux situations sont envisageables pour chaque contrainte j :

- soit $g_j(x^*) = 0$: on dit dans ce cas que la contrainte j est *saturée*,
- soit $g_j(x^*) < 0$.

Comme précédemment, on définit le Lagrangien :

$$\mathcal{L}(x, \Lambda) = f(x) - \Lambda \cdot G(x)$$

Ici, Λ et $G(x) = (g_1, g_2, \dots, g_m)(x)$ sont des vecteurs et \cdot est le produit scalaire.

Conditions de qualification des contraintes : Pour pouvoir utiliser le Lagrangien dans la résolution d'un tel programme d'optimisation, il suffit que l'une des conditions suivantes soit vérifiée :

(a) soit $s \leq m$ le nombre de contraintes saturées à l'optimum x^* . On suppose sans perte de généralité qu'il s'agit des s premières contraintes g_j , $j \in [1..s]$. Si la matrice Jacobienne de ces s fonctions contraintes est de rang s lorsqu'elle est évaluée à l'optimum x^* , alors la condition de qualification des contraintes est vérifiée.

(b) Les fonctions contraintes g_j , $j \in [1..s]$ sont toutes linéaires.

Les conditions du premier ordre que doit vérifier toute solution x^* sont légèrement modifiées par rapport au cas précédent et portent le nom de "conditions de Kuhn-Tucker". Elles ont notamment pour propriété que le multiplicateur λ_j est nul lorsque la contrainte j n'est pas saturée à l'optimum.

Conditions du premier ordre (Kuhn et Tucker) : On suppose que la condition de qualification est vérifiée. Si le vecteur x^* est une solution du programme de maximisation \mathcal{P} , alors il existe un unique vecteur Λ^* appelé *multiplicateur de Lagrange* tel que $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ vérifie les $n + 3m$ conditions suivantes :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x_i}(x^*, \Lambda^*) = 0 & \forall i \in [1..n] \\ \frac{\partial \mathcal{L}}{\partial \lambda_j}(x^*, \Lambda^*) = 0 & \forall j \in [1..m] \\ \lambda_j^* \geq 0 & \forall j \in [1..m] \\ \lambda_j^* g_j(x^*) = 0 & \forall j \in [1..m] \end{cases}$$

Les deux dernières lignes permettent le passage de contraintes en équations à des contraintes en inéquations. Soit la contrainte j est saturée, et on revient à une contrainte en équations ; soit elle est non-saturée, et on l'ignore simplement en choisissant $\lambda_j = 0$.

Conditions du second ordre pour un optimum global : Supposons qu'il existe un x^* qui vérifie les CPO.

- Si f est concave et les g_j sont convexes, alors x^* est un maximum global.
- Si f est convexe et les g_j sont concaves, alors x^* est un minimum global.

4.5.4 Problème dual

Définition 17 (Fonction duale).

$$d(\Lambda) = \begin{cases}] -\infty, +\infty] & \text{si } \Lambda \geq 0 \\ \sup_{x \geq 0} \mathcal{L}(x, \Lambda) & \\ +\infty & \text{sinon} \end{cases} \quad (4.7)$$

D'après les conditions de Kuhn-Tucker, on obtient donc le théorème suivant :

Théorème 9. *Un vecteur Λ^* est un multiplicateur de Lagrange si et seulement si c'est une solution du problème d'optimisation dual :*

$$\min_{\Lambda \geq 0} d(\Lambda) \quad (4.8)$$

4.6 Conclusion

Pour le protocole *Profiterole*, j'ai choisi de représenter les interactions entre stations du réseau par un graphe de conflit dont :

- les sommets correspondent aux stations du réseau,
- il y a une arête entre deux sommets du graphe de conflit si ces deux stations sont à une distance inférieure ou égale à k sauts dans le réseau.

Les contraintes utilisées sont les *contraintes par lignes* évoquées plus haut (section 4.1.3) :

$$\forall n \in N \quad \sum_{i \in V(n)} x_i \leq c$$

où N est l'ensemble des sommets du graphe de conflit, $V(n)$ le voisinage du sommet n dans le graphe de conflit, n compris, x_i est la quantité de bande passante utilisée par le nœud i en émission et c est la capacité du medium radio. Initialement, le but de ce modèle était d'exprimer la contention entre émetteurs à distance de détection de porteuse (mécanisme CSMA). Ce choix modélise donc le partage du medium radio de manière imparfaite puisque i) il ne prend en compte que les contentions entre émetteurs, mais pas les interférences et ii) le voisinage à k sauts est une approximation de la zone de détection de porteuse ou de la zone d'interférences.

Nous avons vu dans la section 4.1 que les contraintes par lignes peuvent être arbitrairement surcontraintes, ce qui pourrait être a priori un inconvénient majeur. Elles ont cependant l'avantage d'être plus adaptées à une implémentation protocolaire distribuée pour deux raisons. Premièrement, le calcul des cliques maximales, s'il est plus simple dans un graphe de conflit de réseau sans fil que dans un graphe quelconque, reste plus complexe que le calcul des voisins à k sauts. La deuxième raison à trait au nombre de contraintes que ces deux modèles impliquent. Avec les contraintes par lignes, on obtient autant de contraintes que de sommets dans le graphe de conflit. Bien que je ne connaisse pas exactement l'évolution du nombre de cliques maximales dans un graphe de conflit de réseau ad hoc, j'ai pu constater lors de mes expériences que celui-ci est toujours plus important que le nombre de sommets dans le graphe de conflit, et semble croître exponentiellement avec le degré du graphe de conflit, résultats consignés dans le premier article qui a été publié sur *Profiterole* [65].

La section 4.2 a montré que les contraintes par cliques sont adaptées aux graphes de conflit des réseaux ad hoc, si on accepte qu'ils sont des Unit Disk Graphs. Mais il est intéressant de remarquer que les contraintes par cliques sont en quelque sorte des contraintes par lignes relâchées. Considérons l'ensemble des voisins à $k/2$ sauts d'une station du réseau. Toutes les stations de cet ensemble sont à k sauts ou moins l'une de l'autre. Ce sont donc des cliques pour l'interaction à k sauts. Ce ne sont pas exactement les cliques maximales, mais avec un peu de réflexion, on se rend vite compte que les cliques dans lesquelles on ne peut pas trouver de centre (i.e. un sommet dont tous les sommets sont tous à moins de $k/2$ sauts de distance) ont peu de chances d'apparaître dans un graphe de conflit de réseau ad hoc, ou différent si peu d'un voisinage à k sauts que cette approximation est minime au regard des approximations de propagation radio. J'ai conscience que ces remarques manquent de rigueur, mais il me semble qu'il était important de montrer pourquoi l'utilisation des contraintes par lignes est intéressante, et remplace avantageusement les contraintes par cliques.

Dans la section 4.3, nous avons vu que la distance en nombre de sauts représente correctement la distance réelle pourvu que la densité de stations soit suffisamment forte. Les résultats théoriques et expérimentaux sur le rapport entre distance de détection de porteuse et distance de communication, exposés en première partie, donnent une valeur comprise entre deux et trois pour ce rapport. Des valeurs pour k de 1 ou 2 semblent donc adaptées, puisqu'elles correspondent respectivement à une prise en compte des interactions à deux et quatre sauts.

Enfin, la section 4.5 sur la relaxation Lagrangienne trouvera elle son utilité dans le chapitre suivant puisque c'est elle qui permet de résoudre le problème d'optimisation des débits des flux de manière distribuée.

Chapitre 5

Description de *Profiterole*

Le monde matériel repose sur l'équilibre, le monde moral sur l'équité.
Victor Hugo.

5.1 Solution théorique

5.1.1 *Profiterole*, un protocole *a la mode*

En plus d'être un dessert français mondialement connu²⁴, *Profiterole* est aussi une proposition de protocole visant à résoudre les problèmes d'équité dans le partage de bande passante dans les réseaux ad hoc multi-sauts. L'appellation de ce protocole suit la mode informaticienne qui consiste à préférer un acronyme accrocheur à une suite imprononçable de lettres. *Profiterole* signifie ainsi :

- PROportional
- FaIr
- raTE allocation
- pRotocOL

***Profiterole* est un protocole de régulation de débit.** Nous avons vu dans les parties précédentes que les inégalités dans les réseaux ad hoc multi-sauts proviennent dans la majorité des cas d'un scénario du type : deux stations sont en concurrence pour l'accès au médium, mais une seule des deux stations le sait. Ainsi, la deuxième station émet en permanence sans savoir qu'elle en gêne une autre (voir les exemples des stations cachées asymétriques ou des trois paires). De nombreux protocoles de régulation de débit nécessitent un retour - *feedback* en anglais - c'est-à-dire un moyen de savoir qu'il est nécessaire de réduire le débit pour laisser de la place aux autres, information qui est difficile à déduire, par exemple, des pertes de paquets, informations utilisées par *TCP*, puisqu'elles ne signifient pas nécessairement la présence de congestion, et peuvent être provoquées par un lien temporairement défectueux. De la même manière, les tailles des files d'attente au niveau MAC, utilisées dans le protocole *DiffQ*, bien qu'elles soient un indicateur plus fiable de la présence de congestion que les pertes de paquets, ne peuvent résoudre les problèmes asymétriques cités plus haut.

***Profiterole* est un protocole de couche 3.** Ce n'est pas le rôle de la couche 3 du modèle OSI (i.e. la couche routage) que de faire la régulation de débit. En principe, c'est plutôt le rôle des couches supérieures. Mais dans le cas des réseaux sans fil, la couche transport (couche 4) ne dispose pas de suffisamment d'informations pour réguler les débits. La couche routage, elle, nous donne des informations sur la topologie du réseau, et sur la présence d'autres flux à proximité, indications utiles pour la régulation. De plus, *Profiterole* est un protocole autonome, dans le sens que, selon les critères OSI, il ne dépend pas - ou

²⁴Rappelons ici que Outre-Atlantique, le terme "a la mode" vu sur une carte des desserts signifie étonnamment "avec une boule de glace à la vanille"

peu - des couches supérieures. Le protocole a été construit pour fonctionner aussi bien avec des protocoles de transport de type TCP qu'avec des protocoles sans acquittement de type UDP. En revanche, il est dans sa version actuelle indissociable des couches inférieures, c'est-à-dire de la couche IEEE 802.11.

Une régulation de type *proportionnelle-équitable*. Le terme *Proportional Fair* fait référence à l'équité proportionnelle, compromis entre maximisation du débit total et égalité parfaite (voir la section 4.4).

Profiterole est simple. C'est une des lignes directrices que nous avons choisies pour l'élaboration de ce protocole. Compte tenu de la complexité de la propagation radio, nous avons décidé de ne pas reposer sur des hypothèses trop contraignantes, et avons privilégié la simplicité, dans le but d'aboutir à un protocole utilisable en pratique.

Un modèle basé sur les nœuds. Je rappelle qu'on modélise les interactions entre stations du réseau par un graphe de conflit dont :

- les sommets correspondent aux stations du réseau,
- il y a une arête entre deux sommets du graphe de conflit si ces deux stations sont à une distance inférieure ou égale à k sauts dans le réseau.

Avec les contraintes

$$\forall n \in N \quad \sum_{i \in V(n)} x_i \leq c$$

où N est l'ensemble des sommets du graphe de conflit, $V(n)$ le voisinage du sommet n dans le graphe de conflit, n compris, x_i est la quantité de bande passante utilisée par le nœud i en émission et c est la capacité du medium radio.

5.1.2 Enoncé du problème d'optimisation

Conventions de notation

- Les scalaires sont notés avec des lettres minuscules.
- Les vecteurs sont notés avec des lettres majuscules, généralement notés horizontalement pour des questions de place : $A = (a_1, a_2, \dots, a_n)$. $(A)_k$, ou a_k est la $k^{\text{ème}}$ composante du vecteur A .
- Les matrices par des lettres de la forme \mathbb{A} . De même, \mathbb{A}_{ik} est le terme de la $i^{\text{ème}}$ ligne, et $k^{\text{ème}}$ colonne.

Notations

On pose :

- n nœuds dans le graphe de conflit, f flux.
- $\Phi = \phi_1, \phi_2, \dots, \phi_n$ le vecteur des débits des flux qui sont transmis dans le réseau.
- $\mathbb{E} \in \mathbb{R}^{n \times f}$ la matrice d'émission des flux, donnant pour chaque flux la liste des stations émettrices de ce flux :

$$\mathbb{E}_{i\phi} = \begin{cases} 1 & \text{si } i \text{ est un émetteur pour le flux } f, \\ 0 & \text{sinon.} \end{cases}$$

- $\mathbb{V} \in \mathbb{R}^{n \times n}$ est la matrice d'adjacence du graphe de conflit :

$$\mathbb{V}_{ij} = \begin{cases} 1 & \text{si les stations } i \text{ et } j \text{ sont à } k \text{ sauts de distance ou moins dans le réseau,} \\ 0 & \text{sinon.} \end{cases}$$

- $C \in \mathbb{R}^n$ est le vecteur de capacité du medium radio, qu'on suppose identique dans l'ensemble du réseau : $C = (c, c, \dots, c)$.²⁵

Alors :

²⁵Cette hypothèse est une contrainte liée au modèle choisi. Bien qu'en pratique, différents liens puissent avoir différentes capacités, on ne peut pas parler de capacité dans une zone géographique, dans le voisinage d'une station. Bien que cela

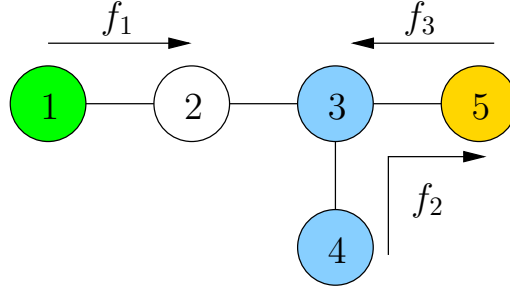


FIG. 5.1 – Un exemple de graphe de conflit. Les flux f_1 et f_3 ont respectivement pour seuls émetteurs les nœuds 1 et 5, ceux du flux f_2 sont les nœuds 3 et 4. Pour réduire la taille de l'exemple, le graphe de conflit ne contient que les interactions à un saut ($k = 1$).

- $\mathbb{E}\Phi$ est un vecteur dont le $i^{\text{ème}}$ terme est le débit émis par le nœud i tous flux confondus,
- $\mathbb{V}\mathbb{E}\Phi$ a pour $i^{\text{ème}}$ terme la somme des émissions dans le voisinage à k sauts de la station i .

Les contraintes par lignes s'écrivent :

$$\mathbb{V}\mathbb{E}\Phi \leq C$$

Cette équation exprime que pour toute station du réseau, la somme des émissions dans son voisinage à k sauts doit être inférieure à la capacité du réseau. D'après le théorème 2 (voir chapitre 4), ces contraintes sont suffisantes pour assurer l'existence d'un ordonnancement des communications tel que deux stations à une distance de k sauts ou moins (ou deux sommets voisins dans le graphe de conflit) n'émettent pas simultanément.

Enfin, on pose :

$$U(\Phi) = \sum_{f \in \Phi} \ln \phi_f$$

U est la fonction objectif assurant l'équité proportionnelle entre les débits des flux.

On a alors le problème d'optimisation suivant :

$$\mathcal{P} \quad \left| \begin{array}{l} \max_{\Phi} U(\Phi) \\ \text{s.c} \quad \mathbb{V}\mathbb{E}\Phi \leq C \end{array} \right. \quad (5.1)$$

5.1.3 Exemple

L'exemple de la figure 5.1 illustre ces équations en fait assez simples. Ici, le réseau contient 5 stations, et 3 flux dont un flux à 2 sauts. Pour cet exemple, le graphe de conflit ne représente que les interactions à un saut ($k = 1$). Le graphe de conflit est donc le graphe de communication du réseau.

$$\mathbb{E} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbb{V} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad \Phi = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix}$$

n'ait pas été fait dans l'implémentation actuelle de *Profiterole*, une solution intéressante à ce problème est d'allouer des durées d'occupation du médium plutôt que des débits. Cependant, d'un point de vue protocolaire, cela suppose que l'on est capable de savoir à quel débit correspond tel ou tel temps d'occupation du médium, c'est à dire de connaître le débit physique du lien sur lequel on émet. Cette possibilité reste malgré tout une piste prometteuse d'amélioration de *Profiterole*.

$\mathbb{E}\Phi$, les émissions totales par nœud :

$$\mathbb{E}\Phi = \begin{pmatrix} \phi_1 \\ 0 \\ \phi_2 \\ \phi_2 \\ \phi_3 \end{pmatrix}$$

Contraintes :

$$\left\{ \begin{array}{ll} (1) & x_1 + x_2 \leq c \\ (2) & x_1 + x_2 + x_3 \leq c \\ (3) & x_2 + x_3 + x_4 + x_5 \leq c \\ (4) & x_3 + x_4 \leq c \\ (5) & x_3 + x_5 \leq c \end{array} \right. \Rightarrow \left\{ \begin{array}{ll} \phi_1 + \phi_2 & \leq c \\ 2\phi_2 + \phi_3 & \leq c \end{array} \right.$$

5.1.4 Utilisation de la relaxation Lagrangienne

Soit $\Lambda \in \mathbb{R}^n$ le vecteur des multiplicateurs de Lagrange : $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$. Le Lagrangien (défini en section 4.5) pour le programme \mathcal{P} est défini par :

$$\mathcal{L} = U(\Phi) + \Lambda^T C - \Lambda^T (\mathbb{V}\mathbb{E}\Phi) \quad (5.2)$$

D'après le théorème 9, le problème 5.1 est équivalent à son problème dual :

$$\min_{\Lambda > 0} \max_{\Phi > 0} \mathcal{L}(\Lambda, \Phi) \quad (5.3)$$

Le problème 5.3 peut être divisé en deux sous-problèmes. Le premier est la maximisation selon Φ et le deuxième est la minimisation selon Λ .

5.1.5 Sous-problème 1 :

$$\max_{\Phi > 0} \mathcal{L}(\Lambda, \Phi)$$

Dans cette section, on suppose Λ fixé, et on s'intéresse à la restriction de \mathcal{L} aux valeurs de Φ , qu'on notera $\mathcal{L}_\Lambda(\Phi)$.

Théorème 10. *Les débits optimaux vérifient :*

$$\forall f \quad \frac{1}{\phi_f} = \pi_f \quad \text{où} \quad \pi_f = (\Lambda^T \mathbb{V}\mathbb{E})_f \quad (5.4)$$

Démonstration. Notons d'abord que \mathcal{L}_Λ est continûment dérivable deux fois et est concave. On suppose qu'aucune des composantes du maximum $\Phi^*(\Lambda)$ de \mathcal{L}_Λ n'est nulle, puisque dans ce cas, $U(\Phi) = -\infty$. Donc le maximum est atteint quand ses dérivées selon les composantes de Φ sont toutes nulles. On obtient le résultat directement à partir de :

$$\forall f \quad \frac{\partial \mathcal{L}}{\partial \phi_f}(\phi_f^*) = 0 \quad \Rightarrow \quad \frac{\partial U}{\partial \phi_f}(\phi_f^*) - (\Lambda^T \mathbb{V}\mathbb{E})_f = 0$$

□

Dans la suite, π_f sera appelé *prix* du flux f . Il peut être écrit comme la somme des prix partiels π_{if} pour chacun des émetteurs i le long de la route du flux f .

$$\pi_f = \sum_i \pi_{if} \quad \text{où} \quad \pi_{if} = (\Lambda^T \mathbb{V})_f = \sum_{j \in V_k(i)} \lambda_j$$

5.1.6 Sous-problème 2 :

$$\min_{\Lambda > 0} \mathcal{L} \quad (5.5)$$

En injectant l'équation 5.4 dans la définition de \mathcal{L} , on obtient :

$$\mathcal{L}(\Lambda, \Phi^*) = - \sum_f \ln(\pi_f) + \Lambda^T C + \text{constante}$$

d'où :

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = C - \sum_f \frac{1}{\pi_f} r_{if} \quad (5.6)$$

où $r_{if} = (\mathbb{V}\mathbb{E})_{if}$ est le nombre d'émetteurs du flux f dans le k -voisinage du nœud n . Cette équation montre que les dérivées partielles du Lagrangien ne dépendent que d'informations locales, i.e π_f et le nombre d'émetteurs dans le voisinage.

5.2 Vers un algorithme distribué

Alors que la formulation originale du problème $\mathcal{P}(5.1)$ laisse supposer qu'il ne peut être résolu que de manière centralisée par une entité qui connaît toutes les données du problème, la seconde formulation (5.5) permet une résolution distribuée. On utilise pour cela la méthode du gradient, ou méthode de la plus grande pente, dont je rappelle ici rapidement les propriétés.

5.2.1 Méthode de la plus grande pente

Soit une fonction $f(x_1, x_2, \dots, x_n)$ de \mathbb{R}^n dans \mathbb{R} , continûment différentiable, à minimiser. La méthode du gradient est une méthode itérative pour trouver un minimum local de cette fonction. On part d'un $z_0 \in \mathbb{R}^n$ quelconque, on calcule le gradient en ce point, puis on se déplace d'un pas α_k dans une direction de descente. La méthode de la plus grande pente consiste à choisir comme direction de descente la direction opposée au gradient :

$$z_{k+1} = z_k - \alpha_k \nabla f(z_k)$$

où ∇ est l'opérateur gradient : $\nabla f(z) = (\frac{\partial f}{\partial x_1}(z), \frac{\partial f}{\partial x_2}(z), \dots, \frac{\partial f}{\partial x_n}(z))$, et $\alpha_k \in \mathbb{R}^+$, le *pas* du gradient.

Variantes de la méthode de plus grande pente Si $\alpha_k = \alpha$ est une constante, on parle de méthode du gradient à pas constant. Il est généralement préférable de choisir à chaque pas k la valeur de α_k qui minimise $f(z_{k+1})$, c'est à dire qu'on cherche le point le plus bas de f dans la direction du gradient. cette méthode est appelée *steepest descent method with line minimization rule*. Enfin, s'il est impossible de calculer le α_k optimal, on peut prendre un pas variable : $\alpha_k = g(k)$ avec $\lim_{k \rightarrow +\infty} \sum_{i=0..k} \alpha_k = +\infty$. Nous reviendrons sur ces différentes méthodes et sur leur applicabilité à notre problème.

5.2.2 Convergence

Théorème 11. *Soit L tel que :*

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \quad \forall x, y \in \mathbb{R}^n \quad (5.7)$$

Et qu'il existe ϵ tel que :

$$\forall k \quad 0 < \epsilon \leq \alpha_k \leq \frac{2 - \epsilon}{L} \quad (5.8)$$

Alors la méthode du gradient converge vers un minimum local de f .

On peut trouver une démonstration rigoureuse de ce résultat dans [9]. La condition 5.7 est appelée condition de continuité de Lipschitz sur ∇f , et assure - grossièrement - que la courbure de f n'est pas plus grande que L dans toutes les directions. La condition 5.8 assure (1) que $f(z_{k+1}) \leq f(z_k)$, c'est à dire que l'algorithme se rapproche à chaque pas de l'optimal et (2) que l'algorithme converge vers un minimum local, ce qui peut ne pas arriver si, par exemple, α_k n'est pas minoré et que $\sum_{i=0..k} \alpha_k < +\infty$.

Intuitivement, si l'on avance à petits pas dans la direction de la plus grande pente, on est assuré de descendre progressivement. Mais si la courbure est importante, alors un grand pas peut nous mener de l'autre côté du minimum, jusqu'à un point plus élevé que le point précédent.

Théorème 12. [9] *Pour toutes les variantes du gradient énoncées ci-dessus, et si les α_k vérifient la condition 5.8, la méthode du gradient a une convergence linéaire qui dépend des deux valeurs propres extrêmes m_k et M_k de la matrice Hessienne de f au point z_k . :*

$$\frac{\|z_{k+1}\|}{\|z_k\|} \leq \max\{|1 - \alpha_k m_k|, |1 - \alpha_k M_k|\}$$

Pour ne pas rentrer dans les détails, je donne une interprétation intuitive de ce théorème. La matrice Hessienne est la version à plusieurs dimensions de la dérivée seconde. Si le rapport entre les valeurs propres M et m est important au voisinage de l'optimal (on parle alors de problème mal conditionné), cela signifie que la fonction f a une forme de *vallée allongée* en ce point : dans une direction, la dérivée seconde est faible - c'est la direction du vecteur propre associé à la valeur propre minimale m ; dans l'autre, la dérivée seconde est importante - vecteur propre associé à M . Dans ce cas, la méthode du gradient peut converger lentement, en avançant à chaque pas en zig-zag comme le ferait une bille dans une gouttière légèrement penchée. On peut résoudre ces problèmes de mauvais conditionnement par des méthodes de gradient conjugué, dans laquelle la direction de descente dépend aussi de la matrice Hessienne, mais nous verrons que ces méthodes ne peuvent être appliquées à notre problème.

5.2.3 Preuves de convergence avec erreur.

Dans le cas de la minimisation de \mathcal{L} , nous avons besoin d'un théorème supplémentaire établi spécifiquement pour notre problème. L'exécution de l'algorithme de minimisation de \mathcal{L} doit se faire de manière distribuée par plusieurs entités *a priori* complètement désynchronisées. L'implémentation de l'algorithme prévoit que chaque station i est responsable du tarif λ_i . On peut montrer que, sous des conditions relativement légères, l'algorithme converge pourvu qu'on choisisse un pas α suffisamment petit.

Théorème 13. *On suppose qu'il existe un entier β tel que à l'instant t , toutes les entités exécutant l'algorithme ont des informations sur leur voisinage ne datant pas de plus de β pas de l'algorithme. Alors il existe ϵ tel que pour tout pas $\alpha < \epsilon$ l'algorithme asynchrone converge vers le minimum de \mathcal{L} .*

La preuve de ce théorème a été reléguée en annexe A.1.

5.2.4 Algorithme de minimisation de \mathcal{L}

L'algorithme de résolution du problème dual (5.3) va consister dans un premier temps à calculer les tarifs λ_i qui minimisent \mathcal{L} avec un algorithme de gradient, puis à en déduire pour chaque flux le débit optimal, qui est l'inverse de π_f , prix qui ne dépend que des tarifs λ_j du k -voisinage des émetteurs du flux f .

A chaque station i est associé un *tarif* λ_i dont il est responsable. L'algorithme est itératif, et à chaque étape t , les tarifs $\lambda_i^{(t)}$ évoluent dans la direction de la plus grande pente de \mathcal{L} proportionnellement à un pas constant : α . Chaque station peut calculer la nouvelle valeur de $\lambda_i^{(t+1)}$ par la formule :

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} - \alpha \cdot \frac{\partial \mathcal{L}}{\partial \lambda_i}(\lambda) \quad \text{où} \quad \frac{\partial \mathcal{L}}{\partial \lambda_i}(\lambda) = C - \sum_{f \in \Phi} \frac{r_{if}}{\pi_f^{(t)}} \quad (5.9)$$

Le prix d'un flux $\pi_f^{(t)}$ est calculé selon la formule :

$$\pi_f^{(t)} = \sum_{i \in 1..n} r_{if} \lambda_i^{(t)} \quad (5.10)$$

Discussion sur la méthode. L'algorithme de gradient à pas constant est réputé pour ses mauvaises propriétés de convergence quand le problème est mal conditionné. Mais les méthodes habituelles pour accélérer la convergence sont difficiles à mettre en oeuvre dans un protocole réel, notamment car elles se heurtent à notre volonté initiale de faire un algorithme aussi distribué que possible. Par souci de complétude, je passe en revue ces différentes méthodes.

Une des principales difficultés du gradient à pas constant est le choix du pas α . Dans la méthode de descente selon la plus grande pente avec la règle de minimisation linéaire, α_k est choisi à chaque étape de sorte à minimiser $f(z_k - \alpha_k \nabla f(z_k))$. Dans notre cas, on vérifie facilement qu'il est nécessaire de connaître toutes les données du problème pour résoudre cette équation.

Une autre manière de contourner le problème est d'utiliser un pas variable. En choisissant un pas initialement grand et décroissant régulièrement, l'algorithme va d'abord converger grossièrement vers l'optimal, puis s'affiner. On évite ainsi le problème de la convergence initiale, qui peut être long dans la méthode à pas constant si α est choisi trop petit. Se pose alors le problème de la synchronisation des différents acteurs participant à l'algorithme : comment faire en sorte qu'ils aient tous le même pas ? Et surtout, cet algorithme nécessite que la valeur du pas soit uniformément réinitialisée quand les données du problème changent. Sachant que les acteurs participant à l'algorithme sont disséminés dans le réseau, comment le détecter ? Et surtout, comment transmettre rapidement l'information à toutes les entités du réseau ? Assurément, ce type d'approche à pas variable implique des communications supplémentaires, une synchronisation du réseau, propriétés dont on ne veut pas pour un algorithme censé être distribué.

Enfin, je me suis également penché sur la méthode du gradient conjugué, qui permet d'accélérer la convergence des programmes mal conditionnés. Ces méthodes sont populaires, mais malheureusement inapplicables à notre problème. Elles font en effet intervenir les dérivées secondes du Lagrangien qui, là encore, ne peuvent être calculées sans connaître toutes les données du problème.

5.3 Implémentation pratique

5.3.1 Le protocole *Profiterole*

Transformer un algorithme en protocole pose un certain nombre de challenges, comme la gestion de l'asynchronisme, la perte de paquets de contrôle ou le surcoût induit par ces derniers.

Nous avons décidé d'intégrer cet algorithme au protocole de routage AODV car nous pouvons tirer parti des paquets Hello diffusés périodiquement (chaque seconde) pour le maintien des tables de voisinage et de routage, ainsi que des paquets *Route Response* - utilisés pour la construction réactive de la route - afin de mémoriser la route retour. Les messages Hello vont contenir les tarifs (λ_n) de ses voisins à $k - 1$ sauts (incluant le tarif du nœud qui envoie le message Hello). Ainsi chaque nœud connaît les tarifs de tous ses voisins à k sauts.

Le prix d'un flux f (Π_f) est calculé par les nœuds se trouvant sur la route empruntée par f . Pour cela, un nouveau paquet de contrôle est utilisé. Un paquet unicast, appelé *price message* est émis régulièrement par le nœud se trouvant juste avant le récepteur et est retransmis sur la route inverse de celle utilisée par f afin de remonter vers la source de ce flux. Saut après saut, le prix total du flux f est calculé selon la formule 5.10. Quand le *price message* arrive à la source, celle-ci adapte son débit d'envoi du flux f selon le prix reçu, par le truchement d'un token bucket. Enfin, les nœuds incluent, dans les messages Hello, les informations sur les prix des flux qui les traversent. En incluant de plus, les prix reçus des voisins à un saut, chaque nœud peut être informé des prix des flux se trouvant dans son k -voisinage, et adapte son tarif en conséquence selon la formule 5.9.

Le surcoût en termes de bande passante induit par ce mécanisme reste faible, bien que l'implémentation actuelle puisse encore être optimisée : il se limite à quelques informations supplémentaires dans les paquets

Hello, envoyés par chaque nœud chaque seconde²⁶, et aux *price messages* eux aussi envoyés sur une base de 1 paquet par flux, par saut, et par seconde. La figure 5.2 résume le fonctionnement de l'algorithme.

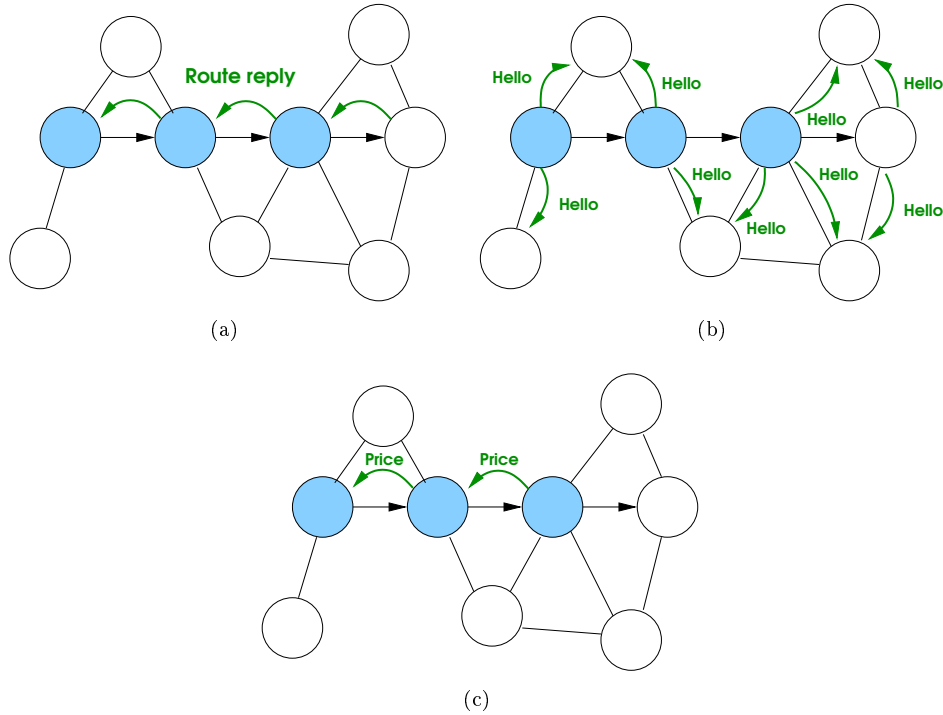


FIG. 5.2 – Implémentation protocolaire de l'algorithme du gradient. Ici, un flux à 3 sauts est représenté par les flèches noires. Les émetteurs du flux sont représentés en bleu. (a) Une route vers la destination vient d'être établie. Le paquet *Route Reply* d'AODV va de la destination à la source et prévient chacun des émetteurs de la route de la présence d'un nouveau flux. Si la route était déjà établie avant l'arrivée du flux, c'est le premier paquet de données qui permettra à chaque émetteur de savoir qu'il l'est (b) Toutes les stations broadcastent régulièrement et de manière asynchrone un paquet Hello informant leur voisinage de leur présence. Ils permettent à chaque station de connaître (1) les tarifs λ_i des stations du k -voisinage, (2) les prix π_f des flux de leur k -voisinage (3) pour chaque flux, le nombre d'émetteurs dans leur k -voisinage (r_{if}). (c) Les émetteurs calculent le prix partiel π_{if} en faisant la somme des tarifs de leur voisinage, puis le prix du flux f (π_f) est recalculé en utilisant le paquet *price message*, et en faisant la somme des prix partiels calculés par chaque émetteur.

5.3.2 Contrôle de flux par token bucket

L'algorithme précédent permet à la source d'un flux de connaître le débit maximal auquel elle peut émettre. Du point de vue de la couche AODV, des paquets proviennent des couches supérieures, et il s'agit donc maintenant de trouver les mécanismes qui permettent de garantir un tel débit. J'ai choisi pour cela un algorithme de type *token bucket*. Le principe est le suivant : on dispose d'un seau, dans lequel on ajoute des jetons à un rythme constant r . S'il n'y a plus de place dans le seau, les jetons à ajouter sont jetés. Chaque jeton correspond à une quantité de bande passante. Ainsi, quand un paquet arrive des couches supérieures, on enlève du seau autant de jetons que ce que ce paquet consomme en termes de bande passante. S'il n'y a pas de jeton, le paquet est immédiatement jeté. L'implémentation de ce token bucket est décrite plus précisément dans l'annexe A.2.

²⁶Rappelons ici que dans 802.11, pour des paquets de faible taille, les temps d'attente d'accès au médium comptent autant que le temps de transfert lui-même en termes d'occupation du médium

L'avantage de ce mécanisme est qu'il permet à la fois de réguler des trafics de type *CBR* (Constant Bit Rate) que des flux *TCP*. Dans le cas de *TCP*, les paquets peuvent parvenir à la couche 3 - dans le pire des cas - sous forme de bursts de paquets de la taille de la fenêtre de congestion, espacés de $2 RTT$ (i.e le temps qu'il faut pour qu'arrivent les acquittements), ce qu'il est possible de faire avec un *token bucket*. La solution naïve qui consiste à laisser passer les paquets à intervalles réguliers - solution de type *leaky bucket* - fonctionnerait mal avec *TCP*. Dans ce type de solution, les paquets doivent être stockés dans une file d'attente jusqu'à ce qu'ils puissent être envoyés, et si la file d'attente dépasse un certain seuil, les paquets sont rejetés. *TCP* réagit alors comme en cas de congestion, c'est à dire trop tard, une fois que tous les paquets qui précèdent le paquet rejeté dans la file d'attente ont été envoyés et qu'aucun acquittement n'arrive pour ce dernier paquet. Dans le cas du *token bucket*, il n'y a pas de file d'attente et donc pas de délai : les paquets sont soit envoyés immédiatement, soit rejetés. Le choix entre *leaky bucket* et *token bucket* revient à un choix entre optimisation du délai et qualité du lissage du débit. J'ai choisi le délai.

5.3.3 Calcul du surcoût du protocole *Profiterole*

Par rapport à l'utilisation du protocole de routage AODV, ou de tout protocole de routage utilisant des paquets Hello réguliers, le protocole *Profiterole* a un coût supplémentaire provenant de deux sources :

- l'ajout dans les paquets Hello des informations de tarif,
- l'utilisation des paquets *Price messages*.

Je montre ici par quelques calculs d'ordre de grandeur que ce surcoût est très faible.

Price messages

Ces messages ne sont utilisés que pour les flux multi-sauts. Tous les T_{price} secondes, le dernier émetteur d'un flux (autrement dit, la station située juste avant la destination d'un flux multi-sauts) envoie un message en direction de la source destiné à calculer récursivement la somme des tarifs dans le k -voisinage de ce flux. Dès qu'elle reçoit la valeur du prix total du flux, la source envoie un nouveau *Price message* dans l'autre direction. Si h est le nombre de sauts que comprend ce flux, et t le coût en temps d'occupation du canal des *price messages*, le surcoût par unité de temps induit par l'utilisation de ces messages s'exprime donc par :

$$\frac{2(h-1)t}{T_{\text{price}}}$$

Ces messages sont de faible taille : ils contiennent les informations d'identification du flux (dans l'implémentation actuelle, l'adresse IP de la source, et celle de la destination), ainsi qu'une valeur de prix. Dans ce cas, les temps de transmission des données, sont négligeables par rapport aux temps d'accès au médium.

Faisons nous une idée avec 802.11b pour un flux à 4 sauts et $T_{\text{price}} = 1\text{s}$. En reprenant les valeurs de calcul de débit de la section 2, un tel paquet utilise $\approx 800\mu\text{s}$ par saut, soit moins de 5ms de temps d'occupation du médium par seconde.

Messages Hello

Le protocole *Profiterole* nécessite une augmentation non négligeable de la taille des paquets Hello, puisqu'ils contiennent :

- les tarifs des stations du $(k-1)$ -voisinage,
- les prix des flux du $(k-1)$ -voisinage.

Pourtant, en termes d'occupation du médium, le surcoût induit par ces messages reste faible. Ici, le seul surcoût provient de l'augmentation de taille des paquets Hello, qui sont de toute façon nécessaires dans AODV. Pourvu que la période d'envoi des paquets Hello, T_{Hello} , soit inchangée par rapport à l'implémentation originale d'AODV, le surcoût se limite donc à l'envoi des données supplémentaires au débit d'envoi des paquets Hello, r . Le surcoût pour un paquet Hello, en supposant une densité de d_s

stations et d_f flux à distance de communication est de l'ordre de :

$$ov_{\text{Hello}} = \frac{1}{r}(k-1)^\beta(d_s + d_f)s_{\text{float}}$$

Note : On approxime le nombre de stations dans le $k-1$ voisinage par $(k-1)^\beta$, où $\beta = 2$ dans le plan, et $\beta = 3$ dans l'espace.

Enfin, chaque seconde, une station entend $\alpha^\beta d_s$ paquets Hello, où α est le rapport entre distance de communication et distance de détection de porteuse, pour une occupation totale du medium de l'ordre de :

$$\frac{\alpha^\beta d_s ov_{\text{Hello}}}{T_{\text{Hello}}}$$

On peut retenir que le surcoût des paquets Hello est en $O((k-1)^\beta d_s^2)$.

Exemple numérique :

Supposons que l'on ait les valeurs suivantes :

d_s	d_f	k	r	α	β	T_{Hello}
10	2	2	1Mbps	2	2	1s

Même en choisissant des valeurs de densité élevée, on obtient un surcoût par paquet Hello d'environ $400\mu s$, soit une occupation du medium dans la zone de détection de porteuse de chaque station de $1.6ms$ par seconde, ce qui reste assez faible pour un réseau aussi dense que celui choisi dans l'exemple. D'autre part, j'ai choisi pour cet exemple une technologie assez ancienne, et le surcoût est d'autant plus négligeable que les débits sont importants. L'utilisation de débits plus élevés, comme ceux de 802.11n devraient permettre d'augmenter la fréquence des paquets Hello et donc la vitesse de convergence de l'algorithme du gradient.

Troisième partie

Evaluation de *Profiterole*

Chapitre 6

Simulations

6.1 Le simulateur NS-2

6.1.1 Performances et réalisme

NS-2 (Network Simulator 2) est un des simulateurs de réseaux filaires et sans fil les plus populaires dans la communauté informatique, notamment car, à la différence d'Opnet [56] ou Qualnet [57], la version commerciale de GlomoSim, il est libre de droits, et son code est accessible et modifiable. Pour que des résultats de simulation soient utiles à la communauté scientifique, il est important que ceux-ci soient vérifiables, reproductibles, et comparables à d'autres approches. À ce titre, la popularité a été pour moi un critère primordial pour le choix du simulateur.

Cependant, NS-2 n'est pas un simulateur très performant. Les concepteurs du simulateur WSNet présentent dans [36] une étude sur le compromis entre les performances d'un simulateur et la précision de ses modèles de propagation radio et de couche physique. Ils montrent que NS-2 est dans l'ensemble assez peu performant, et ne permet pas de réaliser des expériences de grande taille (cent stations au plus, sur une durée d'une centaine de secondes).

Du point de vue du réalisme des simulations, il est notable que l'implémentation de la couche physique de 802.11 dans NS-2 présente de nombreuses inexactitudes (voir par exemple [21]). Cependant, il n'existe pas à l'heure actuelle de simulateur de réseaux sans fil suffisamment fiable pour qu'il puisse se substituer à des expériences en environnement réel, et il semble que NS-2 ne soit pas plus mauvais qu'un autre simulateur de ce point de vue [6]. D'autre part, deux bibliothèques intégrées en 2008 à NS-2.33 proposent un certain nombre d'améliorations : 802.11Ext [21], et dei 802.11 Multirate [24]. Elles n'ont cependant pas été utilisées dans les expériences qui vont suivre, d'abord car je n'ai pas pu optimiser *Profiterole* pour ces extensions, mais aussi car les protocoles de routage tels que AODV, qui souffrent du phénomène de *gray zone*, fonctionnent mal avec ces extensions. Afin de limiter la complexité de l'analyse de *Profiterole*, et de dissocier les problèmes de la couche routage du fonctionnement de l'algorithme, j'ai donc préféré utiliser une version bien connue de NS-2, dont je décris les inexactitudes ci-dessous. Les résultats de simulation sont donc souvent éloignés de la réalité et sont sujet à caution.

6.1.2 Inexactitudes du modèle de propagation radio de NS-2.33

Je présente ici les inexactitudes du modèle de propagation radio de NS-2.33²⁷, en reprenant la liste des hypothèses des réseaux sans fil fréquemment admises, mais pourtant erronées, proposée par Kotz et al. [43] :

1. Le monde est plat. Dans NS-2, il est possible de spécifier des coordonnées en trois dimensions, mais cette troisième coordonnée n'a d'importance que pour la prise en compte des auto-interférences dues

²⁷ A ma connaissance, il s'agit du même modèle que celui utilisé dans la version plus récente de NS, NS-3.

aux rebonds sur le sol, phénomène appelé *multi-path fading*. D'autre part, les simulations faites avec NS-2 ne contiennent pas d'obstacles.

2. La portée de transmission d'une antenne est isotrope. Dans NS-2.33, la puissance reçue ne dépend que de la distance à l'émetteur, et les antennes sont donc supposées omnidirectionnelles et parfaites.

3. Les antennes ont toutes la même portée. 4. Si je t'entends, tu peux m'entendre. Ces propriétés impliquent que tous les liens sont symétriques. Les paquets peuvent donc emprunter la même route à l'aller qu'au retour.

5. Si je t'entends, alors je t'entends parfaitement. NS-2.33 utilise un modèle de réception à seuil. En revanche, les extensions 802.11Ext [21], et de 802.11 Multirate [24] permettent l'utilisation d'un modèle de réception où la probabilité de réception est une fonction du rapport signal sur bruit à l'émetteur.

6. La puissance reçue est une fonction simple de la distance entre émetteur et récepteur.

A cela, il faut ajouter le fait que NS-2 ne permet pas d'observer le phénomène de *gray zone*, car la probabilité de réception ne dépend pas du débit physique auxquels sont envoyés les paquets. Cette dernière remarque a une implication très importante pour le protocole *Profiterole*, puisque les paquets Hello transportant les informations de prix sont envoyés en broadcast, et donc à un débit physique de base. Ils auraient donc une portée plus importante en réalité que dans ces simulations. Dans ces simulations, la valeur de k , c'est-à-dire la moitié de la distance, en nombre de sauts, en deça de laquelle on considère qu'il peut y avoir contention, a été fixée à une valeur de 2. Il est fort probable qu'une valeur de 1 soit plus adaptée si l'on prend en compte la portée des messages Hello.

6.1.3 Paramètres de simulation

Dans toutes les expériences qui suivent, les mêmes paramètres de simulation ont été utilisés. Ils sont décrits dans le tableau 6.1.

6.2 Etude d'un flux

6.2.1 Flux CBR

Cette section permet d'étudier les mécanismes intrinsèques du protocole. On considère un flux UDP/CBR à 4 sauts, tel que représenté en figure 6.1. Dans une telle configuration, les contraintes au niveau de la station 3, ou de la station 2 impliquent que la somme des émissions des quatre émetteurs du flux doit être inférieure à la capacité du medium. Le flux se voit donc attribuer un quart de la capacité, comme on peut le voir sur le graphique des débits (figure 6.2(a)).

Tarifs. La figure 6.2(b) montre l'évolution des tarifs de chaque station pendant les vingt premières secondes de l'expérience. Ces tarifs correspondant aux λ_i , les multiplicateurs de Lagrange associés à chaque contrainte. Après une première phase d'augmentation importante, qui se traduit par un débit réduit dans les premières secondes de l'expérience, les tarifs se mettent brusquement à redescendre. Ce changement brutal de comportement est dû à un délai sur la diffusion du prix total de la route. Le *price message*, dont l'objectif est de calculer le prix total de la route ne parvient pas à la station 1 à cause de collisions dues aux émissions des stations 4 et 5. Les tarifs évoluent donc en fonction du valeur de prix datant de quelques secondes, et continuent donc de grimper jusqu'à ce qu'une valeur de tarif parvienne à la station 1 puis soit diffusée à l'ensemble des stations.

Paramètre	Valeur
Simulateur	NS-2.33
Modèle de propagation	Two Ray Ground
Antennes	omnidirectionnelles
Distance de communication	200 m
Distance de détection de porteuse	400 m
Couche physique	802.11b
Fenêtre de contention	31-1023
Short Retry Limit	4
Long Retry Limit	7
Slot	20 μs
SIFS	10 μs
RTS/CTS	désactivés
Taille des paquets de données	fixe - 1000 octets
Débit physique	11 Mbps
Débit physique de base	1 Mbps
Débit utile théorique d'un flux UDP	5.1 Mbps
Débit utile théorique d'un flux TCP	3.7 Mbps
Paramètres de <i>Profiterole</i>	
Portée des messages Hello (k)	2 sauts
Fréquence des paquets Hello	1 seconde
Fréquence des envoi de messages Price	1 seconde
Fréquence d'actualisation des tarifs	0.1 seconde

TAB. 6.1 – Paramètres de simulation

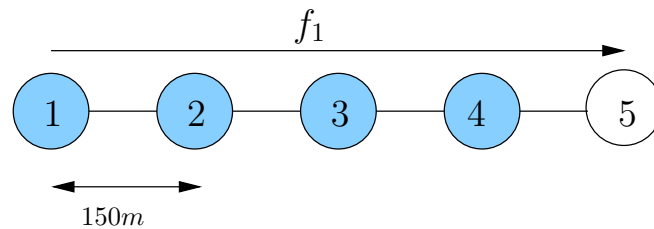
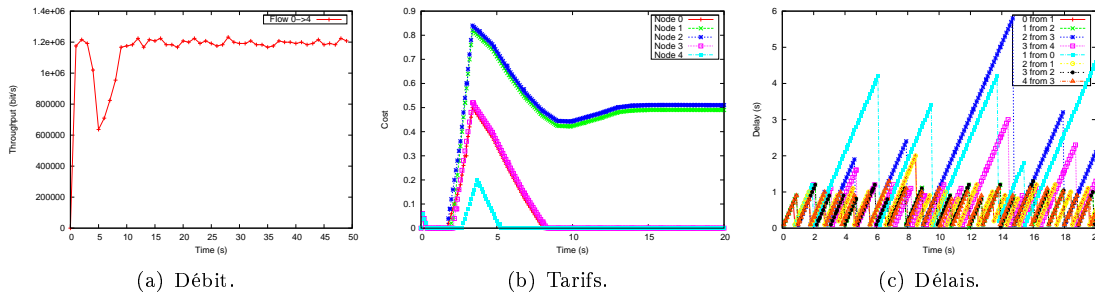


FIG. 6.1 – Flux à quatre sauts. La distance entre les stations est telle que des émissions en provenance de 4 ou 5 peuvent provoquer des interférences en 1. La distance de détection de porteuse est de 400m.

Délais. La figure 6.2(c) donne une idée de l'âge des informations connues par chaque station sur les stations voisines. Pour la majorité des stations, le délai entre le calcul d'un nouveau tarif et sa réception par ses voisins à un saut reste compris entre 0 et 1 seconde. Rappelons que dans l'implémentation originale d'AODV, les paquets Hello, qui permettent de transmettre ces informations, sont envoyés à peu près toutes les secondes. Ce n'est donc pas un résultat surprenant tant qu'aucun paquet Hello n'est perdu. Par opposition, le sort semble s'acharner sur certaines stations, qui ont souvent des informations relativement obsolètes sur un de leurs voisins. Ce phénomène est du à des pertes très fréquentes de paquets Hello sur un lien en particulier (en l'occurrence, des problèmes de stations cachées). Ce type de phénomènes peut nuire à la rapidité de convergence de l'algorithme de gradient. Il est particulièrement apparent dans cet exemple où il n'existe qu'une seule route pour aller d'une station à une autre. Dans des réseaux plus denses, il existe généralement plusieurs routes pour aller d'un point à un autre, ce qui atténue le phénomène.

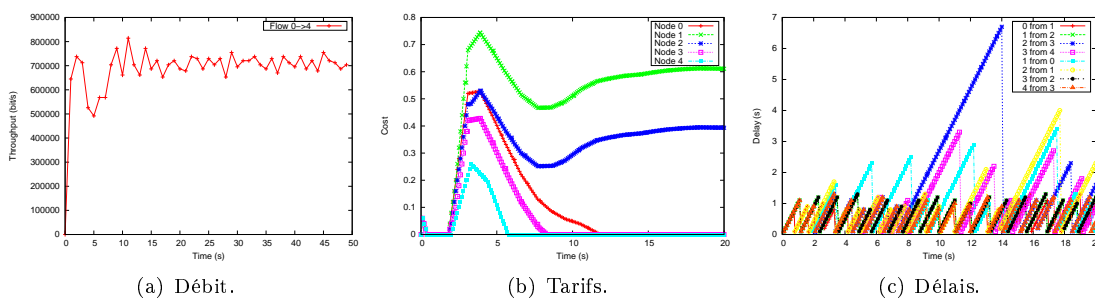
FIG. 6.2 – Evaluation de *Profiterole* sur un flux à 4 sauts - flux UDP.

6.2.2 Flux TCP

Les résultats suivants montrent que *Profiterole* peut aussi être appliqué à des flux TCP. En fonction du type de flux, *Profiterole* calcule l'occupation moyen que l'envoi d'un paquet de ce flux représente. En effet, la bande passante prise par un paquet TCP est plus importante que celle prise par un paquet UDP, puisqu'il faut aussi prendre en compte l'acquittement TCP. Les acquittements consomment une bande passante significative : le débit théorique d'un flux TCP seul avec 802.11b et des paquets de 1000 octets est de 3.7 Mbps.

On peut constater que la régulation de débit par token bucket fonctionne pour TCP. Les variations de débit à court terme que l'on observe sur la figure 6.3(a) ne sont pas dues à une instabilité de l'algorithme - on peut voir que les tarifs sont stables au bout de 20 secondes sur la figure 6.3(b) - mais aux variations de la fenêtre de contention de TCP. Le fait que les tarifs semblent évoluer différemment et vers des valeurs différentes que dans l'expérience précédente n'est pas lié à TCP. A ce sujet, il est intéressant de noter qu'il existe souvent plusieurs multiplicateurs de Lagrange, c'est-à-dire plusieurs tarifs λ_i , optimaux. En revanche, les prix, i.e. la somme des λ_i sur la route de chaque flux, convergent vers une solution unique.

Dans ce cas de figure, *Profiterole* n'apporte pas d'avantage notable par rapport à l'utilisation de 802.11 sans régulation de débit, car (a) tous les liens sont de bonne qualité et (b) il n'y a pas de problèmes de routage.

FIG. 6.3 – Evaluation de *Profiterole* sur un flux à 4 sauts - flux TCP.

6.2.3 Flux à nombreux sauts

Quand le nombre de sauts augmente, la convergence de l'algorithme devient plus lente. On peut observer en figure 6.4 les résultats obtenus pour un flux TCP comprenant 20 stations - et donc 19 sauts. Les oscillations des tarifs et du prix du flux que l'on observe dans les figures 6.4(b) et 6.4(c) sont dues au temps nécessaire à l'algorithme pour calculer la somme des tarifs sur l'ensemble de la route puis pour

diffuser cette valeur à l'ensemble des stations voisines du flux. Cette expérience soulève le problème du choix du pas du gradient de l'algorithme : s'il était choisi un peu plus grand, l'algorithme pourrait ne pas converger dans ce genre d'expériences. La longueur des flux est la première source de problèmes de convergence dans *Profiterole*.

On constate sur la figure des débits (figure 6.4(a)) que dans ce cas, la régulation faite par *Profiterole* n'a aucun impact sur le débit du flux. En effet, le débit maximal que TCP parvient à atteindre sur une telle topologie est plus faible que la valeur théorique calculée par *Profiterole*, égale au cinquième de la capacité, i.e. $0.74Mbps$. Ici, le paramètre limitant est la taille de la fenêtre de congestion de TCP, limitée à 20 dans nos expériences.

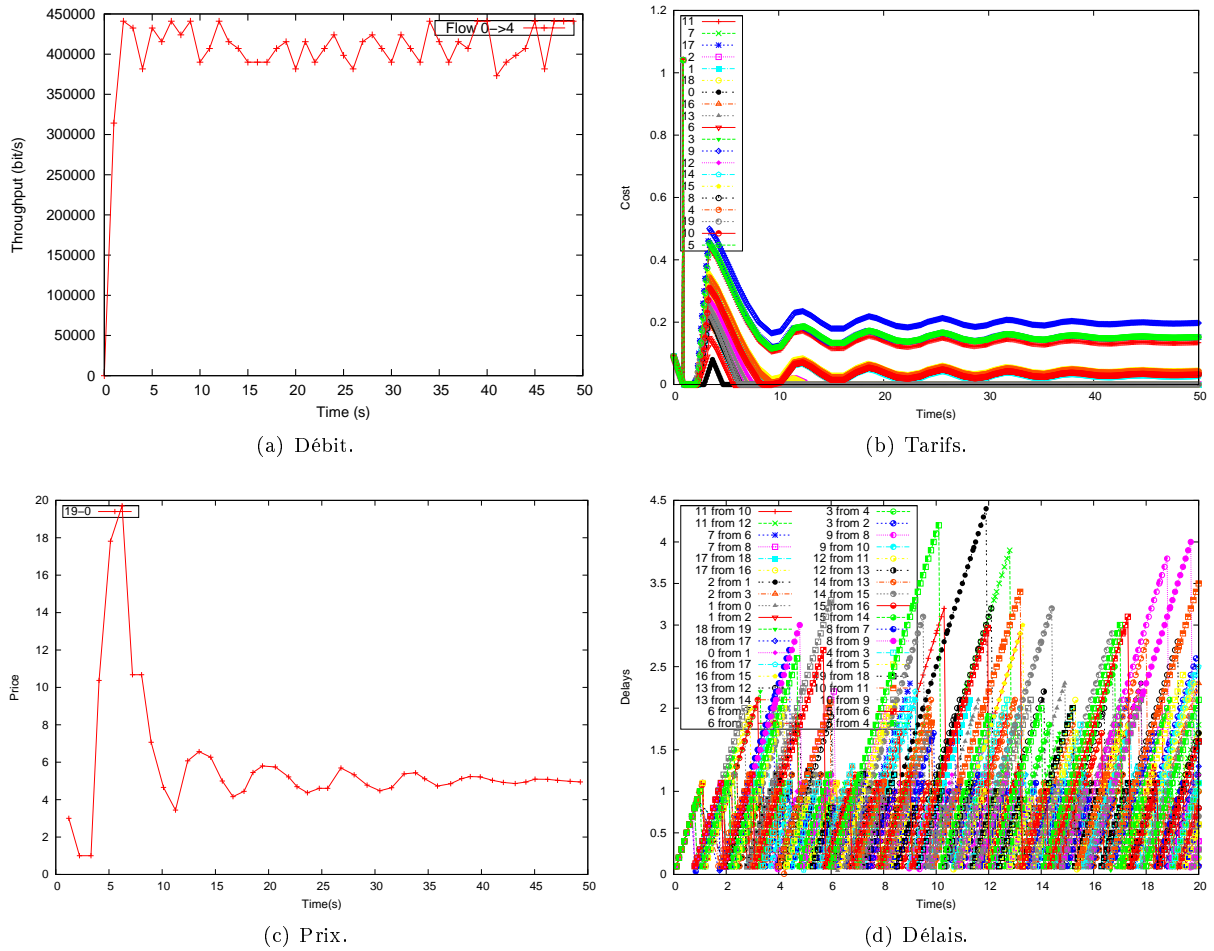


FIG. 6.4 – Flux TCP à 20 stations (19 sauts).

6.3 Station cachée asymétrique

Considérons maintenant le scénario de la station cachée asymétrique, dont la topologie est représentée en figure 6.5. Dans cette configuration, le protocole *Profiterole*, s'il permet de réduire les inégalités de partage, ne peut supprimer complètement les problèmes de collisions.

Dans la configuration des stations cachées asymétriques (voir 2.3.1 pour plus de détails), deux flux sont en compétition pour l'accès au médium, le flux 3-4 occupe toute la bande passante car ses émissions provoquent des collisions au niveau du récepteur du flux 0-1. C'est la situation que l'on observe pendant les 8 premières secondes de l'expérience en figure 6.6(a). Après une phase de convergence (secondes 8 à

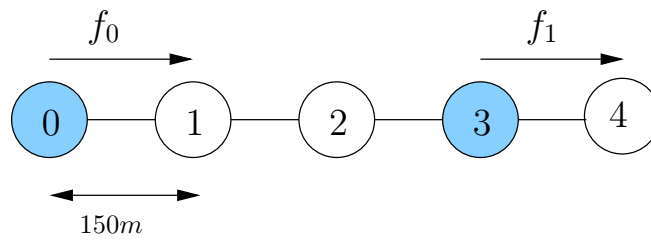


FIG. 6.5 – Stations cachées asymétriques.

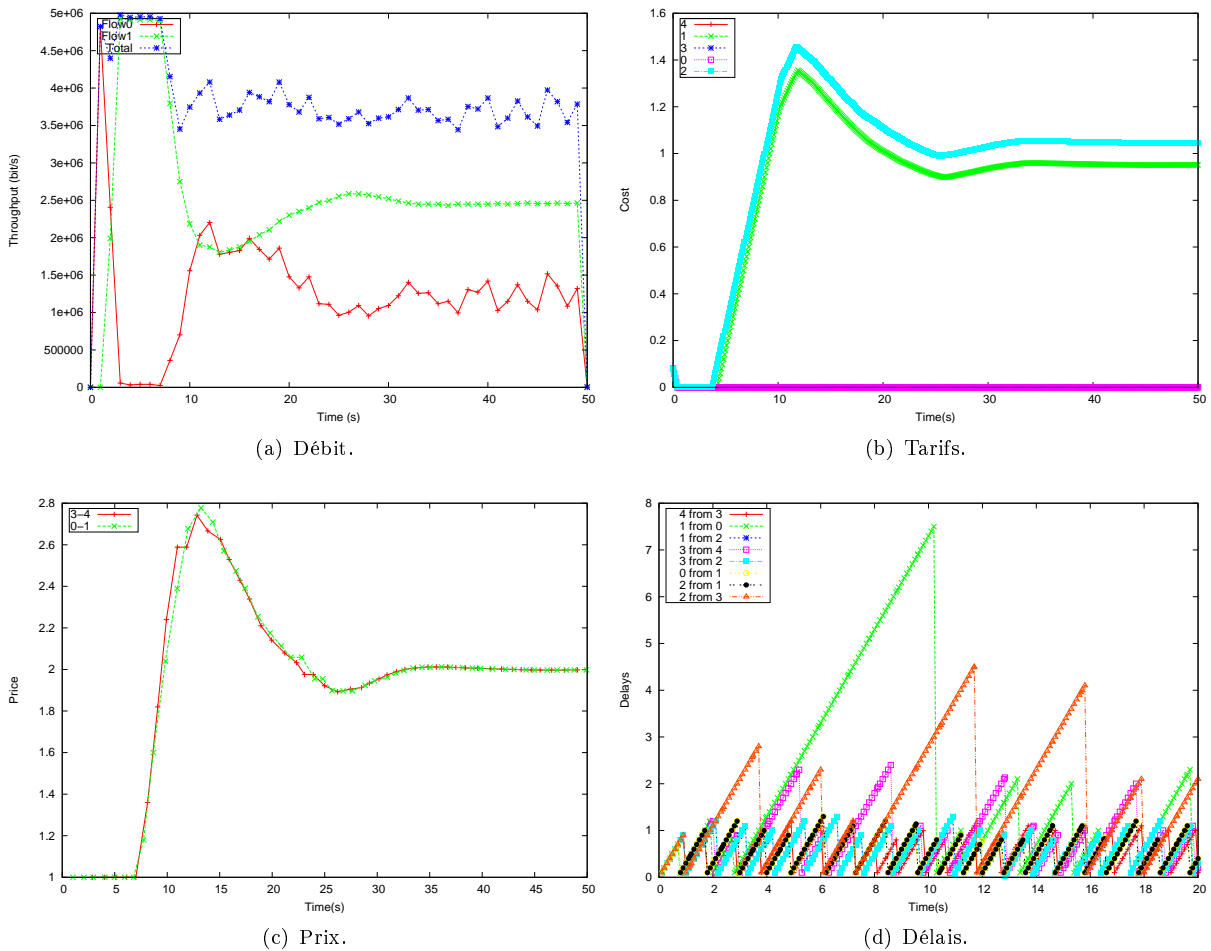


FIG. 6.6 – Résultats de l'évaluation pour les stations cachées asymétriques.

	802.11b	802.11g	802.11n ^a
SIFS	10 μs	10 μs	16 μs
Slot time	20 μs	9 ou 20 μs	9 μs
Fenêtre de contention (nombre de slots)	31-1023	15-1023	15-1023
Débit d'envoi maximal	11Mbps	54 Mbps	540Mbps

^adonnées pour la couche physique WWiSE, qui est une des propositions pour la couche physique 802.11n

FIG. 6.7 – Temps d'attente dans quelques variantes de 802.11

20), les prix des flux se stabilisent à une valeur de 2, ce qui signifie que chaque flux se voit allouer la moitié de la capacité.

La figure 6.6(d) représente pour chaque station l'âge des informations de coût qu'elle a sur chacun de ses voisins. Comme des paquets Hello sont envoyés toutes les secondes environ, et les informations de tarif sont actualisées tous les dixièmes de seconde, cet âge ne dépasse pas environ 1.1s tant que les paquets Hello sont reçus correctement. On constate dans cette configuration que certaines stations souffrent de délais élevés : la station 1 ne reçoit pas de message Hello de 0 pendant plus de 7s, ce qui veut dire que 7 paquets Hello consécutifs ont été perdus lors de collisions avec des émissions provenant probablement de 3. Ce délai, que l'on observe souvent quand il n'existe qu'une seule route d'une station à une autre, peut être parfois problématique pour la vitesse de convergence du protocole, comme c'est le cas ici. Puisque 0 est la source d'un flux, il est responsable de la diffusion du prix de ce flux, qui ne peut donc avoir lieu jusqu'à la seconde 12, quand un paquet Hello parvient à la station 1. Les tarifs continuent donc d'évoluer avec une valeur de prix obsolète jusqu'à cet instant. Par la suite, la régulation du flux 3-4 permet de réduire le risque de collision, et ainsi faciliter la transmission des messages Hello. A cause de ce genre de comportement, on est contraint de garder un pas du gradient plus faible pour éviter les divergences de l'algorithme de descente (voir résultats théoriques de convergence en section 5.2.3). On peut conclure de cette expérience que l'usage d'un canal séparé pour les messages Hello, ou d'une méthode pour fiabiliser le transport de ces messages, accélérerait grandement la convergence.

Dans cette configuration, la régulation de débit ne résout que partiellement le problème, puisque des collisions ont toujours lieu : alors que le flux 3-4 obtient la moitié de la capacité (2.5Mbps), la moitié des paquets envoyés par 0 ne sont pas reçus par 1. La proportion de collisions que la station 1 subit semble être exactement la proportion de bande passante occupée par 3, et on est tenté d'en tirer une conclusion erronée : la relation qui relie ces deux grandeurs n'est pas aussi simple. D'une part, 3 n'occupe pas physiquement le médium pendant 50 % du temps, puisque le temps d'envoi d'un paquet est composé en majeure partie de temps d'attente (essentiellement le décompte de la fenêtre de contention). D'autre part, il y a collision en 1 si, à un instant quelconque de la transmission de 0 à 1, la station 3 démarre une nouvelle transmission. Enfin, à chaque tentative de retransmission d'un paquet ayant subi une collision, la fenêtre de contention double de taille, et les temps d'attente sont donc allongés.

Il semble possible que ce phénomène s'atténue avec les nouvelles variantes de 802.11. En effet, dans 802.11g et plus récemment, 802.11n, la tendance est à une réduction relative des temps d'envoi de données par rapport aux temps d'attente, essentiellement pour des raisons de compatibilité avec les anciennes versions, comme le montre le tableau 6.7. Bien qu'il y ait quelques petites différences entre les mécanismes d'accès au médium entre les variantes de 802.11, il est raisonnable de comparer leurs temps d'attente à partir de la fenêtre de contention et de la taille du slot, qui sont semblables, alors que les temps de transmission sont réduits d'un facteur 50 entre 802.11b et 802.11n. Ainsi, puisque les temps d'attente prennent une place de plus en plus importante, la probabilité de collision baisse, et le phénomène mentionné précédemment pourrait devenir moins significatif.

Nombre de stations	25
hline Taille des simulations	400 × 400
Type de flux	Backlogged / UDP / CBR
Nombre de simulations	20
Durée d'une simulation	50s

TAB. 6.2 – Paramètres de la simulation de grands réseaux

	Débit moyen (kb/s)	Indice de Jain moyen
Sans allocation de débit	647.33	0.67
Avec <i>Profiterole</i>	650.26	0.81

TAB. 6.3 – Débits et Equité pour la simulation de grands réseaux

6.4 Performances en débit sur de grands réseaux

Les séries de simulations qui suivent sont des simulations aléatoires, permettant ainsi de montrer les performances du protocole dans des situations moins spécifiques, et comprenant davantage de stations. Pour chacune de ces simulations, on place 25 stations aléatoirement dans un carré de taille $400m \times 400m$. Quatre flux UDP/CBR traversent ce réseau. La source et la destination de chacun de ces flux est elle aussi choisie aléatoirement. Sachant que la distance de communication dans cette simulation est de 160m, la plupart de ces flux sont donc multi-sauts. Pour chaque simulation, on compare les performances de *Profiterole* avec celles de 802.11 sans régulation de débit. Cette simulation a été répétée 20 fois en faisant varier les flux et la position des stations. Le tableau 6.2 résume les différents paramètres de ces simulations.

Au début d'une simulation, chaque source tente d'émettre à la capacité maximale, puis notre protocole régule progressivement les débits. Nous étudions les débits des flux en moyenne sur les 10 dernières secondes de la simulation, afin de s'assurer que l'algorithme de régulation ait convergé au moment de la mesure²⁸. Nous utilisons l'indice de Jain [58] pour mesurer l'équité des allocations obtenues. Pour N flux de débits respectifs $(x_i)_{i \in [1..N]}$, cet indice est défini ainsi :

$$\frac{(\sum x_i)^2}{N * \sum x_i^2} \quad (6.1)$$

L'indice de Jain peut varier entre $\frac{1}{N}$ (un seul flux obtient l'ensemble de la capacité) à 1 (tous les flux ont le même débit).

Les résultats de ces simulations sont résumés dans le tableau 6.3, ainsi que dans la figure 6.8. Bien que le débit moyen reste inchangé, *Profiterole* augmente significativement l'équité des allocations. *Profiterole* est même plus efficace en termes d'utilisation du réseau. En observant les simulations en détail, on constate que ce sont les flux multi-sauts qui sont le plus défavorisés avec 802.11, alors que *Profiterole* a tendance à réduire les débits des flux à un saut pour assurer aux flux multi-sauts un débit raisonnable. On pourrait donc s'attendre à ce que la moyenne des débits soit plus faible avec *Profiterole*, puisqu'un flux à k sauts occupe k fois plus de bande passante qu'un flux à 1 saut. C'est essentiellement un effet collatéral de la régulation de bande passante : en évitant la saturation du canal radio, on amoindrit les risques que les paquets de routage soient perdus. Avec 802.11, de nombreux flux ne parviennent pas à trouver de route vers leur destination et affichent donc un débit nul.

La figure 6.9 donne la distribution cumulative des flux selon leur débit pour les 80 flux simulés. L'allocation proportionnelle équitable réduit significativement le nombre de flux pénalisés : la proportion de flux émettant à moins de 50kbps passe de 20% à 5%. Dans les simulations avec *Profiterole*, un seul flux n'avait pas du tout de bande passante : il s'agissait d'un flux pour lequel l'émetteur était tout simplement hors de portée de communication du reste du réseau.

²⁸Dans la majorité des simulations, une vingtaine de secondes sont nécessaires pour atteindre l'allocation optimale.

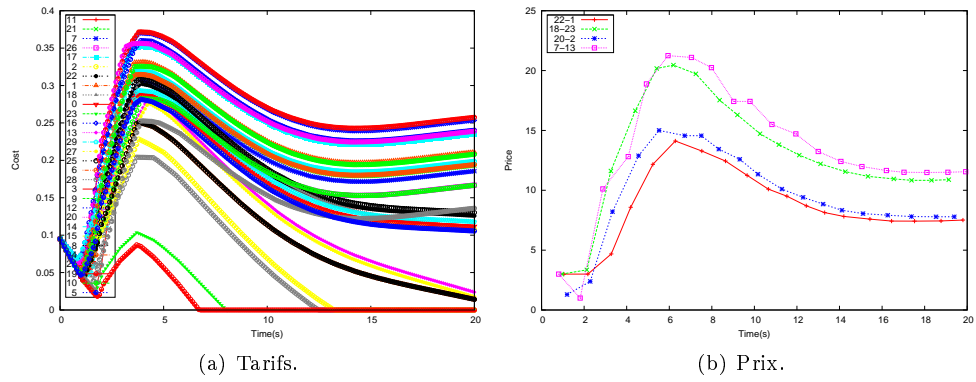


FIG. 6.8 – Tarifs et prix lors d’une exécution de la simulation 4 flux - 25 stations.

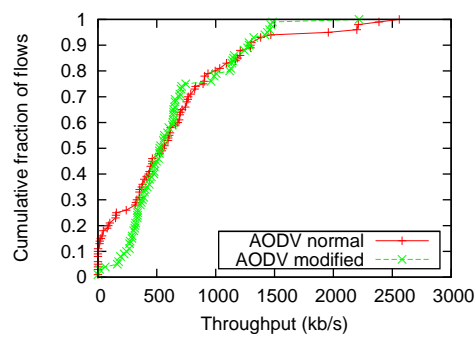


FIG. 6.9 – Fonction cumulative de distribution des débits des flux pour un réseau de 25 stations

6.5 Comparaison avec un protocole de couche MAC

(tiré de [33])

Dans cette partie nous donnons les résultats de simulation dans lesquels nous comparons le protocole *Profiterole* au protocole *DiffQ*. Les paramètres de simulation sont présentés dans le Tableau 6.4. Certains paramètres de *DiffQ* n'étant pas donnés dans [68], nous les avons choisis à partir des recommandations des auteurs. Il n'est donc pas impossible que ces paramètres ne soient pas optimaux. Les flux sont des flux UDP lorsque ce n'est pas spécifié.

Simulation	Simulateur	ns-2.33
	Temps	120s
	Taille	500m × 500m
	Mobilité	Aucune
	Portée Com.	150m
	Portée Interf.	300m
Protocole MAC	802.11b	Paramètres Classiques
	File d'attente	Taille fixe (50 paquets)
Protocole Routage/Regulation	AODV	Paramètres Classiques
	DiffQ	[68]
		AIMD : $\alpha = 2/3$, $\beta = 1kbps$
		Queue_Thresh = 10
<i>Profiterole</i>	[65]	
Trafic	CBR / UDP	Capacité max
	FTP / TCP	Capacité max
	Départ des flux	Aléatoire

TAB. 6.4 – Résumé des paramètres de simulation.

Dans cette section, nous comparons l'efficacité des deux approches du point de vue du débit obtenu. En effet, les deux approches sont fondamentalement différentes dans la manière de réguler le débit de chaque flux dans un réseau.

Dans *DiffQ*, une auto-régulation de chaque flux à la source combinée à l'ajout de priorités au niveau de l'accès au médium tend à réguler les débits des flux de manière équitable. La régulation est faite au niveau MAC, à partir d'informations locales : sur chaque route, chaque nœud informe régulièrement le nœud précédent sur la taille de sa file d'attente; pour chaque nœud, c'est la différence entre la taille de la file d'attente locale et celle du saut suivant, appelée *diff*, qui va déterminer la priorité MAC d'un paquet à envoyer : plus cette différence est grande, plus le paquet est prioritaire. Cette simple politique d'ordonnancement liée à la taille des files d'attente permet ainsi à une station ayant une file d'attente saturée (pour une destination donnée) d'avoir une plus grande priorité au niveau MAC lui permettant de purger cette file d'attente. Notons que dans *DiffQ*, il y a une file d'attente par destination et le choix de la file d'attente à purger sur un nœud se fait aussi en fonction de la variable *diff*. Dans *Profiterole*, contrairement à *DiffQ*, on utilise des informations de plus haut niveau (niveau routage) et des informations sur le voisinage à 2 sauts pour réaliser cette régulation.

Les résultats de cette section sont divisés en trois parties. Dans les deux premières, nous comparons les deux protocoles sur deux scénarios spécifiques. Ces deux scénarios sont les trois paires et une chaîne de communication augmentée de flux saut par saut. Dans le scénario des trois paires 6.10(a), nous voulons faire apparaître un problème d'équité dû à l'accès asymétrique du médium entre la paire centrale et les paires extérieures. Les résultats présentés dans la section 6.5.1 montrent que l'approche proposée dans *DiffQ* ne résoud pas le problème. En revanche, *Profiterole* exhibe des résultats équitables. Dans le deuxième scénario testé (chaîne de communication 6.10(b)), nous voulons faire apparaître le problème des interférences intra-flux. Les résultats présentés dans la section 6.5.2 montrent que *DiffQ* et *Profiterole* obtiennent de bons résultats. Dans la troisième partie donnée en section 6.5.3, nous présentons des résultats sur des scénarios aléatoires. Ces résultats montrent que l'approche proposée dans *Profiterole* permet d'avoir des résultats plus équitables comparés aux résultats obtenus avec *DiffQ*. De plus, tous les scénarios que nous avons testés ici montrent qu'une approche d'auto-régulation, couplée avec une priorité au niveau MAC, comme proposée dans *DiffQ*, ne permet pas d'obtenir des débits réguliers, stables et un partage équitable de la bande passante, contrairement à *Profiterole* basé sur un partage explicite de la

bande passante. *Profiterole* semble donc être un bon point de départ pour obtenir de la qualité de service dans les réseaux ad hoc. Ces résultats sont commentés plus en détails dans les sections suivantes.

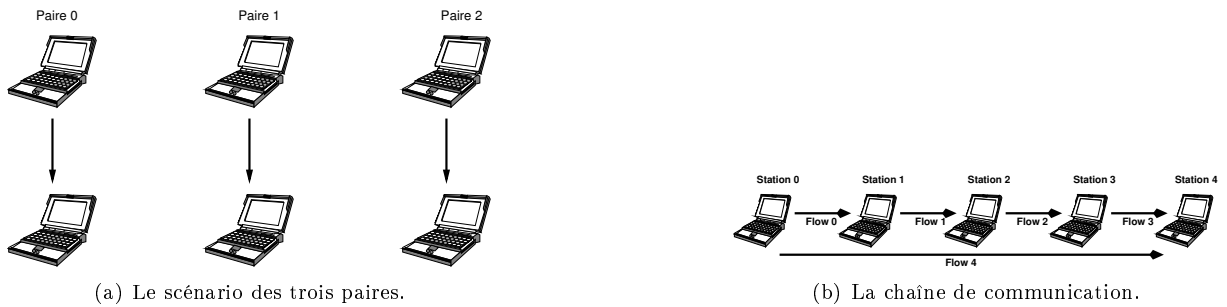


FIG. 6.10 – Scénario étudiés.

6.5.1 Trois paires

Le scénario des trois paires Fig. 6.10(a) a été étudié dans [18]. Dans ce scénario, la paire centrale, ici Paire B, est en concurrence pour l'accès au médium avec les paires extérieures (A et C). Cette asymétrie dans l'accès au médium provoque une famine au niveau de la bande passante pour la paire centrale (illustrée dans la figure 6.11(a)). Pour éviter cette famine, un partage de la bande passante est nécessaire.

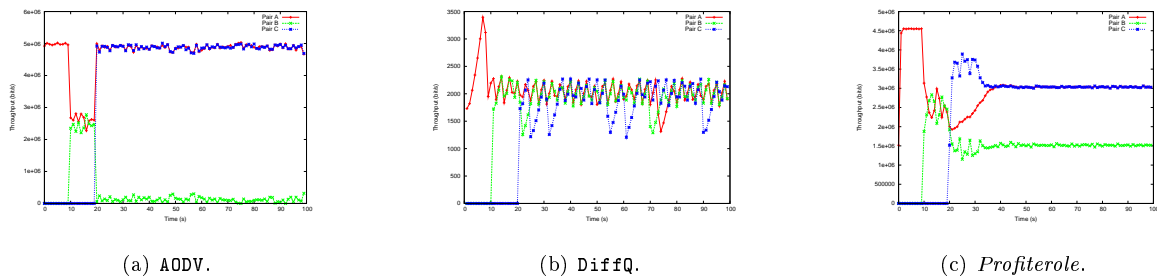


FIG. 6.11 – Evolution du débit en fonction du temps dans le scénario des trois paires.

La figure 6.11 présente les résultats de simulation pour *Profiterole*, *DiffQ* et *AODV* sans régulation de débit. Ces figures tracent l'évolution des débits de chaque paire en fonction du temps. Ces figures montrent que l'approche proposée dans *Profiterole* est meilleure d'un point de vue de l'équité dans le partage de la bande passante. Il est à noter ici que l'obtention de cette équité se fait au détriment d'une baisse du débit global. La figure 6.11(b) montre que *DiffQ* affiche de meilleures performances en termes d'équité qu'*AODV*. Ce comportement résulte du fait que l'auto-régulation proposée dans *DiffQ* augmente la priorité au niveau MAC pour les stations ayant une file d'attente pleine (et donc la paire centrale ici). En revanche, *DiffQ* réduit aussi le débit à la source quand la file d'attente de celle-ci est pleine (ce qui est le cas ici pour la paire centrale). C'est la concurrence de ces deux actions qui permet à *DiffQ* d'avoir de meilleurs résultats qu'*AODV*, mais moins bons que *Profiterole*. En effet, dans *Profiterole* les débits de chaque flux sont calculés explicitement ce qui permet d'avoir ce partage de la bande passante.

Commentaires pour la courbe de *Profiterole* : Dans la première partie de la courbe (de 0 à 10 secondes), la paire A est seule, et dispose de l'ensemble de la capacité. Dans la deuxième partie (10 à 20 secondes), les paires A et B partagent le médium. On constate pendant la première seconde de cette partie une augmentation rapide de la bande passante allouée à la paire B. Dans la version de *AODV* sans régulation, cette augmentation est immédiate. Ici, elle prend une à deux secondes, le temps que l'algorithme de régulation constate l'arrivée de la paire B et partage le médium équitablement. Dans

une troisième partie, à l'arrivée de la troisième paire, on constate que la paire B voit son débit réduire brutalement, et la paire C obtenir plus de bande passante que la paire A . En effet, à cet instant, les paires A et B sont régulées à la moitié de la capacité : du point de vue de la paire B , elle est donc en concurrence avec les paires A et C la moitié du temps, et se trouve alors en situation de famine, et en concurrence avec la paire C seule pendant l'autre moitié. Et effectivement, on constate qu'à cet instant, la paire B obtient le quart de la bande passante.

6.5.2 Chaîne de communication

Dans le scénario de la chaîne de communication, présenté dans la figure 6.10(b), nous voulons faire apparaître le problème des interférences intra-flux. Ici, nous avons ajouté des communications saut par saut pour accroître les interférences sur le flux entre la station 0 et la station 4.

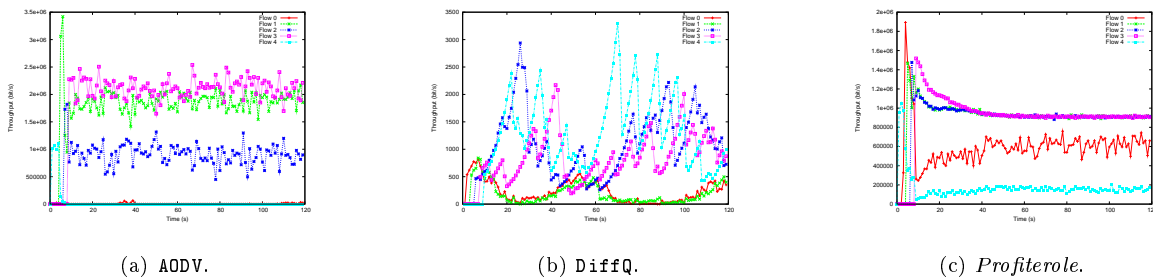


FIG. 6.12 – Evolution du débit en fonction du temps dans le scénario de la chaîne de communication.

La figure 6.12 montre ici encore que l'approche que nous proposons dans *Profiterole* permet un partage de la bande passante plus équitable. En effet, les courbes de la figure 6.12(c) montrent que chaque flux obtient un débit (constant) au cours du temps. De plus, le débit le plus faible est toujours strictement supérieur à 0 ce qui dénote une absence de famine au cours du temps pour tous les flux. Ces résultats sont dus au partage de la bande passante effectué dans notre protocole. Nous voyons aussi sur ces figures que les débits obtenus avec *DiffQ* sont instables. Il est important de noter ici que contrairement à *Profiterole* et *AODV*, le flux le plus avantage dans *DiffQ* est le flux 4 (entre la station 0 et la station 4). Cet avantage est dicté par la politique d'ordonnancement de *DiffQ* qui donne une préférence aux flux ayant un grand nombre de paquets dans la file d'attente et dans ce scénario au flux consommant le plus de ressources dans le réseau. Notons aussi que pour *AODV*, le flux 0 (entre les stations 0 et 1) et le flux 4 (entre les stations 0 et 4) obtiennent un débit nul dû aux collisions provoquées au niveau de la station 1 (en réception) par les transmissions de la station 3 (ceci correspond au problème des stations cachées asymétriques [10]).

6.5.3 Scénarios aléatoires

Dans cette section, nous avons effectué des simulations aléatoires. Nous considérons un réseau de taille $500m \times 500m$ avec 30 nœuds placés aléatoirement. Dans ces simulations, les flux commencent à des instants différents choisis aléatoirement entre $[0; 100]s$. Ainsi, les débit moyens des flux sont calculés à partir de 100s. Les autres paramètres sont conservés tels quels. Notons que les mêmes couples sources/destinations sont utilisés pour les simulations des différents protocoles.

6.5.4 Flux UDP

Nous choisissons aléatoirement 5 sources et 5 destinations avec des flux UDP/CBR. La figure 6.13 montre les résultats de ces simulations. Il faut noter ici que l'évaluation de performance sur un réseau aléatoire est complexe. En effet dû à l'aléa, il est difficile de connaître les interactions qui peuvent potentiellement expliquer certains phénomènes.

Nous pouvons voir sur les résultats présentés dans la figure 6.13 que, comme pour la chaîne de communication, *Profiterole* permet d'avoir plus d'équité au niveau des débits. En effet, nous pouvons voir

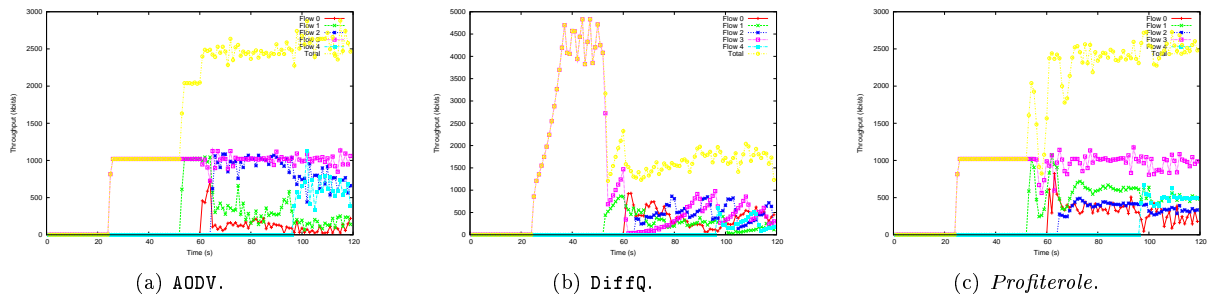


FIG. 6.13 – Evolution du débit en fonction du temps sur un scénario aléatoire (30 noeuds, 5 flux, udp).

que le débit le plus faible obtient un débit autour de 250kbps avec *Profiterole*. Pour *DiffQ*, le débit le plus faible se trouve autour de 120kbps et pour *AODV* il est de 70kbps . Nous pouvons constater sur ces figures que le débit global obtenu par *AODV* est le plus élevé des trois protocoles (2601.40kbps) au détriment de l'équité comme soulevé précédemment. Le débit global obtenu par *Profiterole* est de 2500.22kbps et celui obtenu avec *DiffQ* est de 1722.18kbps . Ces valeurs de débit global et de débit minimum indiquent que *Profiterole* permet d'obtenir une allocation de débit équitable et efficace en même temps. Avec *DiffQ*, cette équité se fait au détriment de l'efficacité, tandis qu'*AODV* privilégie l'efficacité au détriment de l'équité. Cette remarque se vérifie bien en observant les flux 3 et 0 pour les différents protocoles. En effet, *Profiterole* et *AODV* arrivent à maintenir le débit du flux 3 autour des 1000kbps alors qu'avec *DiffQ* ce flux n'obtient qu'environ 470kbps . En revanche, avec *DiffQ* le flux 0 obtient 410kbps alors qu'avec *Profiterole* et *AODV* ce flux obtient beaucoup moins de débit. Ce comportement est dû à la politique d'ordonnancement de *DiffQ* qui tend à purger les files d'attente les plus grandes. Les résultats sur chaque flux sont résumés dans le tableau 6.5.

	Flux 0	Flux 1	Flux 2	Flux 3	Flux 4	Total
<i>AODV</i>	72.62	159.93	724.20	1006.53	638.11	2601.40
<i>DiffQ</i>	415.75	124.04	456.14	477.36	248.88	1722.18
<i>Profiterole</i>	255.0	489.6	323.54	949.00	483.07	2500.22

TAB. 6.5 – Résumé des débits obtenus par simulation. Scénario aléatoire (30 noeuds, 5 flux, udp).

6.5.5 Flux TCP

Dans cette section, nous choisissons aléatoirement 2 sources et 2 destinations avec des flux TCP et un trafic FTP. La figure 6.14 montrent les résultats de ces simulations. Ici nous avons choisi de ne considérer que 2 flux car les flux TCP impliquent, dans le sens inverse, les flux des acquittements.

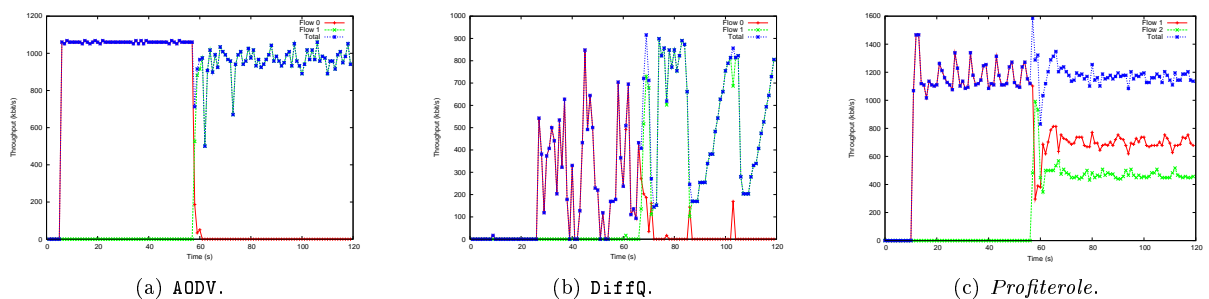


FIG. 6.14 – Evolution du débit en fonction du temps sur un scénario aléatoire (30 noeuds, 5 flux, udp)

Les résultats de la figure 6.14 montrent que les performances de *Profiterole* sont meilleures que celles

de DiffQ et de AODV. Ces résultats montrent que l'approche dans laquelle la bande passante est allouée au flux (données et acquittements) permet d'obtenir des résultats plus équitables. En effet, sur cette figure nous voyons que le débit minimum obtenu vaut environ 460kbps pour *Profiterole*, 0kbps pour AODV et 9kbps pour DiffQ²⁹. Il faut noter que de par sa conception, DiffQ ne peut pas obtenir de bonnes performances avec TCP du fait que les acquittements ont difficilement la priorité dans l'ordonnancement mais aussi à cause du fait que DiffQ propose une régulation supplémentaire et similaire à celle de TCP. Nous voyons aussi sur la figure 6.14 que le débit global obtenu par AODV est proche celui obtenu par *Profiterole* et supérieur à celui de DiffQ grâce à la famine provoqué sur le second flux. Un résumé des débits obtenus pour les différents flux est donné dans le tableau 6.6.

	Flux 0	Flux 1	Total
AODV	0.0	971.00	971.00
DiffQ	8.48	535.08	543.56
<i>Profiterole</i>	464.7	697.48	1162.18

TAB. 6.6 – Résumé des débit obtenus par simulations. Scénario aléatoire (30 nœuds, 2 flux, tcp).

6.5.6 Conclusion

Les résultats obtenus sur NS-2 montrent un bon comportement de *Profiterole* sur différents scénarios et ce aussi bien avec des flux UDP qu'avec des flux TCP. Nous avons comparé *Profiterole* au protocole DiffQ, nouvelle solution de contrôle de congestion pour les réseaux sans fil multisaits proposée récemment. Les résultats montrent que *Profiterole* est plus équitable, plus efficace et plus stable que DiffQ. L'auto-régulation de débit basée sur les différences des tailles des files d'attentes entre nœuds voisins et sur une priorisation au niveau MAC, telle que proposée dans DiffQ, a certes l'avantage d'être simple, mais ne permet pas de tendre vers un partage équitable du médium radio comme le réalise *Profiterole*. La régulation explicite de *Profiterole* est, pour ces scénarios, plus efficace que la régulation implicite de DiffQ.

²⁹les débits sont calculés à partir de 100s.

Chapitre 7

Équité et fonctions d'utilité

tiré de [28]

7.1 Préambule

Ce chapitre un peu à part est un travail conjoint avec Manos Dramitinos sur la notion d'équité, et sur la satisfaction de l'utilisateur d'un réseau. Au delà des formules mathématiques - dont le moins qu'on puisse dire est qu'elles ne sont pas toujours intuitives - ce travail apporte un regard nouveau sur les méthodes d'évaluation des protocoles réseau. Nous avons vu dans les sections précédentes que le débit moyen d'un ensemble de flux renseigne assez mal sur la qualité de cette allocation de débit. En effet, dans les problèmes d'optimisation convexes, l'allocation qui offre le meilleur débit moyen est aussi la moins équitable possible (voir section 4.4). Le débit moyen est donc un indicateur peu fiable de la qualité d'un protocole.

Mais cette remarque néglige la dimension temporelle. Dans un réseau, la demande change en permanence : des flux arrivent dans le réseau, d'autres se terminent. Et la mobilité intrinsèque des réseaux sans fil accentue ce phénomène. Dans ce contexte, on peut être tenté de se demander s'il est justifié, à tout instant, de sacrifier un peu d'efficacité pour plus d'équité : si, au cours de la vie d'un flux, les conditions évoluent, on peut raisonnablement supposer que celles-ci seront tantôt favorables, tantôt défavorables, mais que dans l'ensemble, elles s'équilibreront. Ainsi, en maximisant l'utilisation du réseau - et donc en ne cherchant pas l'équité instantanée - on maximisera le débit moyen de chaque flux. Tout ceci devrait être plus clair avec l'exemple des trois paires, vu en section 2.3.2. Imaginons un flux dans un réseau sans fil très mobile qui au cours de sa vie, occupe successivement les places des paires A, B et C dans l'exemple des trois paires. La figure 7.1 montre le débit moyen que ce flux aura en fonction du type d'allocation instantané choisi. Il est maintenant évident que l'allocation *instantanée* qui maximise le débit total est aussi celle qui maximise le débit moyen de ce flux, mais il est plus facile de se représenter ce résultat contre-intuitif avec un exemple. Pour un flux dont c'est le débit à long terme qui compte (comme par exemple pour un transfert de fichier), on préférera donc cette dernière approche. Mais qu'en est-il pour un flux de streaming video ?

Le travail qui suit est une tentative de modéliser la satisfaction d'un utilisateur d'un réseau. Il montre notamment que les propriétés d'équité du protocole Profiterole ont un impact positif sur la satisfaction

	Paire A	Paire B	Paire C	Débit moyen du flux mobile
Allocation max-min	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Allocation proportionnelle équitable	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{5}{9}$
Maximisation du débit total	1	0	1	$\frac{2}{3}$

FIG. 7.1 – Débit moyen selon les types d'allocation dans le scénario des trois paires

des utilisateurs, notamment en assurant un débit plus constant.

7.2 Introduction

La majorité des solutions équitables proposées pour les réseaux ad hoc sont évaluées en fonction de leurs performances sur des scénarios statiques : chaque scénario est évalué sur une topologie donnée (elle peut être aléatoire) où des paires émetteur-récepteur fixées communiquent durant l'ensemble de la simulation selon un certain modèle de trafic. Les paramètres étudiés peuvent être le débit par flux, le débit total, la distribution des débits, le produit des débits, ou l'indice de Jain, et ces métriques sont évaluées en moyenne sur toute la durée de la simulation ou sur plusieurs exécutions de la simulation. Parfois, l'équité à court-terme est évaluée en utilisant par exemple l'indice de Jain moyen sur de plus courtes durées.

Mais sur une durée plus longue, les types de trafic et/ou la topologie du réseau sont susceptibles de changer. Ainsi, il est vraisemblable que certains flux qui sont pénalisés à un instant donné de la simulation ne rencontrent pas de difficultés à d'autres moments. De plus, les valeurs moyennes temporelles, que ce soit sur le débit ou sur un indice d'équité quelconque, ne renseignent pas sur la volatilité des débits au cours du temps, ni sur l'impact des solutions proposées sur un certain nombre d'indicateurs de qualité de service, tels que le délai, ou la gigue. Les études sur l'équité à court terme donnent une idée des délais perçus par les nœuds du réseau pour accéder au médium, mais on ne peut pas en déduire leur distribution au cours du temps. Ces métriques donnent donc une mesure de l'utilisation du réseau, mais ne reflètent pas précisément le ressenti des utilisateurs quant à la qualité d'un flux. Pourtant, les utilisateurs sont très sensibles à la qualité de service et indifférents aux protocoles réseaux utilisés et à leur optimisation. Ainsi, nous pensons qu'il est utile d'évaluer l'équité des protocoles des réseaux sans fil à un niveau plus haut, et de proposer une métrique qui reflète la qualité du service fourni par le réseau aux différents flux selon un certain schéma d'équité. C'est ici que se situe l'originalité de notre approche.

Il est donc naturel d'introduire des *fonctions d'utilité* dont l'objectif est de fournir une base commune pour comparer les performances de schémas d'équité dont les objectifs de maximisation sont fondamentalement différents (comme par exemple, l'équité max-min, l'équité proportionnelle, etc.). Dans ce chapitre, nous proposons un cadre d'évaluation pour évaluer les performances des protocoles équitables des réseaux sans fil. Il repose sur des fonctions d'utilité à *mémoire* propres à chaque type de service. La métrique de performance que nous utilisons est le *bénéfice commun*³⁰, la somme des fonctions d'utilité de tous les utilisateurs. Nous pensons que contrairement aux méthodes habituelles pour l'évaluation des protocoles équitables des réseaux sans fil, cette méthodologie reflète précisément la satisfaction des utilisateurs à l'égard d'un certain schéma d'équité. Pour finir, nous donnons des résultats de simulation auxquels nous appliquons cette méthodologie.

Le reste de ce chapitre s'organise de la manière suivante : dans la section 7.3, nous présentons un état de l'art et introduisons notre cadre d'évaluation. La section 7.4 est dédiée à une classification des flux selon leur sensibilité à divers paramètres de qualité de service ; d'autre part, nous y définissons les fonctions d'utilité correspondant à ces services, et discutons de leurs propriétés. Dans l'avant dernière section de ce chapitre, nous démontrons que ce cadre est utile, en comparant les performances du protocole de *Profiterole* avec celles du standard IEEE 802.11, et donnons ainsi des résultats expérimentaux.

7.3 Etat de l'art - notre approche

7.3.1 Etat de l'art

Les travaux qui proposent des solutions aux problèmes d'équité dans les réseaux ad hoc font généralement reposer leurs évaluations de performance sur les débits des flux (voir par exemple [7, 59]). La plupart des scénarios testés sont statiques, ce qui signifie que la topologie est donnée et que les paires de nœuds émetteur/récepteur sont fixés et ne changent pas au cours de la simulation. Pour éviter les biais dus aux particularités des simulations, les débits des flux sont calculés sur la moyenne de plusieurs exécutions

³⁰traduction peut être maladroite du concept de *social welfare*, dans la théorie des jeux.

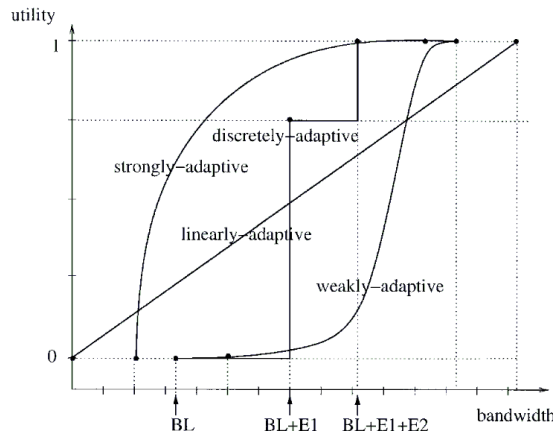


FIG. 7.2 – Fonctions d'utilité typiques. Source : [46].

de cette simulation. Même quand un intervalle de confiance est donné, ces moyennes ne donnent aucune indication sur les fluctuations de ces débits à plus court terme. Les études utilisant l'équité à court terme comme celle de [8] permettent d'obtenir davantage d'informations sur les délais perçus par les stations : en l'occurrence, les auteurs de [8] calculent la moyenne de l'indice de Jain sur de courtes fenêtres de temps. Mais une fois de plus, les valeurs moyennes, ou même une distribution des valeurs, ne permettent pas d'observer les fluctuations des débits, fluctuations qui affectent grandement la satisfaction des utilisateurs pour une grande variété de services.

Depuis les travaux fondateurs de John von Neumann et Oscar Morgenstern [66], les fonctions d'utilité ont été très largement utilisées en économie, pour la prise de décision, ou en théorie des jeux (voir [23, 63, 67] et leurs références). De nombreuses recherches ont été menées pour tenter d'appliquer les fonctions d'utilité à la mesure de la satisfaction des utilisateurs dans un réseau sans fil [45, 46, 52, 62]. Les fonctions d'utilité utilisées dans ces travaux, représentées en figure 7.2, quantifient le bénéfice obtenu par un utilisateur en fonction du débit qu'on lui alloue à un instant donné. Ces valeurs d'utilité instantanées ne sont pas corrélées, et si le débit varie, les valeurs d'utilité varient elles aussi quelles qu'aient été leurs valeurs précédemment. Cette approche néglige donc l'impact des interruptions de service et du délai, qui est un paramètre important de la qualité de service perçue par un utilisateur sur une plus longue durée.

7.3.2 Notre approche

Nous pensons que déterminer l'impact des variations des débits au cours du temps dans les schémas d'équité ad hoc sur la qualité perçue par l'utilisateur, mais aussi sur le *bénéfice commun* est un problème ouvert important. Les fonctions d'utilité à mémoire sont une extension des fonctions d'utilité classiques permettant d'exprimer la satisfaction obtenue à long terme pour un flux dont la qualité varie à court terme. L'unité de temps exprimant le court-terme est désormais appelée *slot*. Pour chaque flux, on calcule le nombre de paquets reçus durant ce slot, et donc le débit moyen associé, dont on déduit une valeur d'utilité instantanée. La durée d'un slot dépend du type de service, puisque chaque service est plus ou moins tolérant aux dégradations de qualité momentanées.

Le principal mérite des fonctions d'utilité à mémoire est que plusieurs paramètres de qualité comme le vecteur des débits instantanés, les délais, et/ou la quantité de ressources allouées ont un impact sur les *utilités marginales corrélées*. Le terme "utilité marginale" fait référence à l'utilité obtenue à chaque slot d'une session. Ces utilités permettent donc de quantifier la satisfaction instantanée d'un utilisateur. Ce type de fonction d'utilité a été proposé pour la première fois dans [29] pour l'allocation de ressource par enchères dans les réseaux UMTS, puis dans d'autres domaines [30, 49]. La version complète de l'article dont est issu ce chapitre [28] fournit des explications détaillées sur les différences entre les fonctions d'utilité que nous proposons ici et celles de [29].

Dans ce chapitre, nous proposons une architecture en trois couches, comme représenté en figure 7.3. Nous utilisons une classification des services reposant sur les paramètres de QoS qui ont une importance

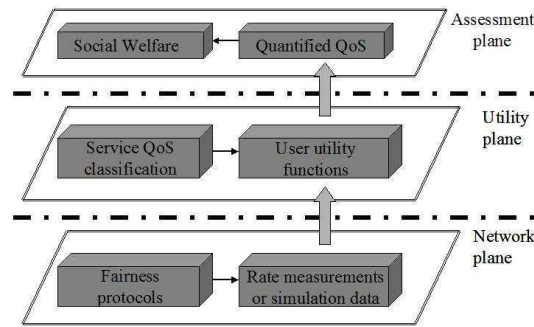


FIG. 7.3 – Structure de notre méthode d'évaluation à base de fonctions d'utilité.

pour ces services, de sorte à définir une fonction d'utilité et une valeur de slot pour chaque classe de service. Ces fonctions d'utilité sont additives, et donc définies comme la somme des utilités marginales obtenues à chaque slot. Les valeurs de ces utilités marginales sont *corrélées*, i.e. elles ne sont pas indépendantes, et selon le type de service et la manière dont le réseau ad hoc sert l'utilisateur, elles varient pour exprimer la satisfaction (ou l'insatisfaction) de l'utilisateur quant aux variations de la qualité du service.

Afin de comparer les performances de divers protocoles ad hoc équitables, le vecteur des débits des flux est calculé slot par slot, soit par des mesures réelles ou des simulations - il s'agit de la couche Réseau³¹. Puis celui-ci est transformé en vecteurs d'utilité - couche Utilité³². Enfin, la couche d'Évaluation³³ quantifie la satisfaction obtenue pour chaque flux, ainsi que le bénéfice commun, qui mesure les performances du protocole étudié. Le bénéfice commun est une métrique très souvent utilisée pour mesurer la satisfaction d'une population dans son ensemble.

Nous prétendons que cette approche est nouvelle et peut servir à la communauté scientifique des réseaux ad hoc comme un outil d'évaluation de performance des protocoles des réseaux ad hoc. En l'appliquant au protocole *Profiterole*, nous démontrons qu'elle permet de détecter de mauvaises performances du standard IEEE 802.11 là où l'étude du débit moyen ne les fait pas apparaître.

7.4 La couche Utilité

Cette section présente la couche Utilité, qui est le cœur du cadre d'évaluation de performance que nous proposons.

7.4.1 Services et QoS

Avant de commencer à définir les fonctions d'utilité, nous commençons par définir les différents types de services, ainsi que leurs besoins en termes de qualité de service. Le projet 3GPP a désormais fourni un travail important dans le contexte des réseaux UMTS sur l'identification des paramètres importants de QoS et sur leur classification. En particulier le rapport TS 23.107 [1] définit quatre classes de QoS : *conversational*, *streaming*, *interactive* et *background*.

Les classes *conversational* et *streaming* appartiennent toutes deux aux services peu tolérants au délai, tels que la voix, ou la vidéo/audio temps réel. Les paramètres importants de QoS pour ces classes sont le *débit minimal* et le *délai*. En particulier, la tolérance maximale au délai pour les services de voix est de 100msec , alors qu'elle est de 250msec pour les applications de streaming [1]. Les classes *interactive* et *background* correspondent à des applications sensibles au débit, comme le téléchargement, ou le web browsing³⁴, pour lesquels c'est la quantité totale de données reçues qui est importante.

Dans ce travail, nous utilisons ces classes de QoS et leurs attributs respectifs pour définir nos fonctions d'utilité.

³¹Network Plane

³²Utility Plane

³³Assessment Plane

³⁴je n'ai pas osé *butinage* ici

7.4.2 Définition des fonctions d'utilité

Sans perte de généralité, nous supposons que l'utilité $u_{s,i}$ qu'obtient l'utilisateur i du service s est la somme des utilités marginales obtenues à chaque slot t à partir des vecteurs des débits alloués dans le passé $\langle x_i^{(1)}, \dots, x_i^{(t)} \rangle$. Ainsi,

$$u_{s,i}(x_i^{(1)}, \dots, x_i^{(t_{s,i})}) = \sum_{t=1}^{t_{s,i}} v_{s,i}^{(t)}(x_i^{(1)}, \dots, x_i^{(t)})$$

Dans cette formule, $t_{s,i}$ est la durée de la session de l'utilisateur du service, et $x_i^{(t)}$ est le débit qui lui est alloué au slot t . En fonction de la classe de chaque service, il nous faut définir une forme appropriée pour $v_{s,i}^{(t)}(x_i^{(1)}, \dots, x_i^{(t)})$. Pour des questions de lisibilité, nous noterons $v_{s,i}^{(t)}(x_i^{(1)}, \dots, x_i^{(t)})$ par $v_{s,i}^{(t)}$.

Classe streaming

Les services de la classe *streaming*, tels que la video ou l'audio en temps réel, sont sensibles au délai et au débit. Pour ces services, nous supposons qu'il y a un débit minimum r_0 en dessous duquel la qualité du service est inacceptable pour l'utilisateur. Cette hypothèse est vérifiée par des travaux sur la QoS subjective dans les réseaux sans fil [70, 42, 14].

Nous définissons $v_{s,i}^{(t)}$ pour cette classe sous la forme :

$$v_{s,i}^{(t)} = \begin{cases} [v_0(s, i, t) + \Delta V_s \cdot f(x_i^{(t)}, tp_i)] \cdot \alpha_s^{d_i} & , \text{ si } x_i^{(t)} \geq r_0 \\ 0 & , \text{ si } x_i^{(t)} < r_0 \end{cases}$$

L'utilité marginale obtenue avec un débit constant de r_0 est v_0 ³⁵. Ainsi, si la valeur économique - et donc le prix qu'un utilisateur est prêt à payer - d'un service de streaming de débit r_0 et de durée T est notée wtp , alors $v_0 = \frac{wtp}{T}$. Tout débit inférieur à r_0 aboutit à une utilité marginale égale à zéro. Cependant, l'utilité marginale est constituée de deux parties : la première partie reflète la satisfaction pour un utilisateur à qui on alloue un débit d'au moins r_0 , alors que la deuxième donne la satisfaction si le débit est supérieur à r_0 .

La première partie $v_0(s, i, t)$ est définie par :

$$v_0(s, i, t) = \begin{cases} v_0 & , \text{ si } t = 0 \\ \frac{1}{\sqrt{d_i}} \cdot v_0 & , \text{ si } d_i > 0 \\ \min\{v_0, v_0(s, i, t-1) + \beta \cdot v_0\} & , \text{ si } d_i = 0 \end{cases}$$

où $\beta \in [0, 1]$ et d_i est la distance entre le slot actuel et le dernier slot pendant lequel l'utilisateur a eu un débit d'au moins r_0 . $v_0(s, i, t)$ baisse brutalement en cas d'interruption de service, puis remonte progressivement si le service atteint le débit minimum selon un mécanisme AIMD³⁶. Ainsi, l'utilité marginale a une mémoire des dégradations de qualité durant les précédents slots.

Toute quantité additionnelle de bande passante induit une augmentation de $\Delta V_s \cdot f(x_i^{(t)}, tp_i)$ qui dépend à la fois de la quantité de bande passante $x_i^{(t)}$ allouée, et du type d'utilisateur tp_i . Selon son adaptabilité, celui-ci peut être dans l'une des trois catégories suivantes : (a) par seuils³⁷, (b) linéaire³⁸, ou (c) fortement adaptatif³⁹ (voir figure 7.2). ΔV_s correspond à la satisfaction supplémentaire pour un utilisateur à qui on alloue l'ensemble de la capacité du canal C , et $f(\cdot)$ est une fonction croissante de la bande passante dont la forme dépend du type d'utilisateur et dont les valeurs varient entre 0 et 1. Cette

³⁵ Pour r_0 and v_0 , nous omettons le s pour des raisons de clarté.

³⁶ Additive increase / Multiplicative Decrease

³⁷ discretely-adaptive sur la figure

³⁸ linearly-adaptive

³⁹ strongly-adaptive

fonction permet de représenter le bénéfice qu'un utilisateur tire d'une augmentation supplémentaire de la bande passante. C est la capacité théorique maximale qu'un flux à un saut peut atteindre. Pour la définition de f , nous utilisons des fonctions standard que l'on trouve dans [46, 52] :

$$f(x_i^{(t)}, tp_i) = \begin{cases} \frac{x_i^{(t)} - r_0}{C - r_0} & , \text{ si } tp_i \text{ est linéairement adaptatif} \\ \frac{\log(x_i^{(t)} - r_0)}{\log(C - r_0)} & , \text{ si } tp_i \text{ fortement adaptatif} \end{cases}$$

Pour finir, pour des utilisateurs de la catégorie adaptatifs-discrets, $f(x_i^{(t)}, tp_i)$ est une fonction en escalier⁴⁰.

L'utilité marginale (constituée de $v_0(s, i, t)$ et $V_s \cdot f(x_i^{(t)}, tp_i)$) est aussi multipliée par un coefficient $\alpha_s^{d_i}$ où $\alpha_s \in [0, 1]$. Ainsi, si le service est interrompu fréquemment (i.e on observe souvent $d_i > 0$), une grande quantité d'utilités marginales seront réduites par ce coefficient, de sorte à refléter l'impact d'interruptions de service fréquentes (i.e. la video se bloque). Notez que $\alpha_s^{d_i}$ reflète le fait qu'un service momentanément après une longue interruption de service a peu de valeur, alors que le terme $v_0(s, i, t)$ fluctue sur une échelle de temps plus grande. Pour résumer, $v_0(s, i, t)$ reflète mieux le débit moyen alloué à l'application, alors que $\alpha_s^{d_i}$ reflète la fréquence des interruptions de service.

Comme cela a déjà été expliqué plutôt, la valeur du slot, i.e. la durée pendant laquelle le débit moyen est calculé est de 250msec . Ainsi, pour une session de 60sec , on obtient un vecteur de 240 valeurs de débits, chacune étant calculée sur une durée de 250msec , puis ces valeurs permettent d'obtenir 240 valeurs d'utilité marginales.

Pour conclure, cette définition de l'utilité reflète l'insatisfaction résultant à la fois (a) de débits faibles, avec une utilité marginale de zéro, et une réduction de la valeur de $v_0(s, i, t)$, et (b) de la durée et de la fréquence des interruptions de service qui affectent le coefficient $\alpha_s^{d_i}$ juste après une interruption de service. D'autre part, il est intéressant de noter que l'augmentation d'utilité provoquée par un débit supérieur à r_0 est variable selon le type d'utilisateur.

Classe Conversational

Comme l'explique la section 7.4.2, les services de la classe *conversational* ont les mêmes critères de QoS que ceux de la classe *streaming*. La différence réside dans l'impact des interruptions de service, qui génèrent une dégradation de la qualité de service plus importante et plus rapide. Ceci est reflété dans la fonction d'utilité par une réduction plus rapide de l'utilité marginale $v_0(s, i, t)$. $v_0(s, i, t)$ est donc défini comme suit :

$$v_0(s, i, t) = \begin{cases} v_0 & , \text{ si } t = 0 \\ \frac{1}{d_i} \cdot v_0 & , \text{ si } d_i > 1 \\ \min\{v_0, v_0(s, i, t - 1) + \beta \cdot v_0\} & , \text{ si } d_i = 0 \end{cases}$$

Nous rappelons au lecteur que dans cette classe, la durée du slot est de 100msec .

Classes Background et Interactive

Les classes *background* et *interactive* contiennent les services sensibles au débit, tels que le téléchargement de données. Les utilités marginales sont donc définies comme le produit du débit alloué à chaque slot par un coefficient constant de normalisation $v_0 : v_{s,i}^{(t)} = v_0 \cdot x_i^{(t)}$. Pour une durée de slot de 1s , v_0 doit représenter la valeur associée à une allocation de 1kb/s .

⁴⁰ce type de fonction est censée représenter les utilisateurs de service de type Skype, où l'application adapte la qualité du son ou de la video en fonction de la quantité de bande passante qu'elle obtient. Cette adaptation se fait par seuils.

7.4.3 Discussion

Les définitions de fonctions d'utilité contenues dans cette section ne sont évidemment pas uniques. Mais leur définition exacte et les paramètres qui ont été choisis ont peu d'importance tant que (a) ces définitions sont rationnelles et reflètent l'impact des paramètres importants de QoS sur la satisfaction de l'utilisateur et (b) on utilise les mêmes définitions de fonction d'utilité pour comparer différents protocoles. Ainsi, la définition précise d'une fonction d'utilité, ainsi que les valeurs de qualité de service obtenues en injectant les débits obtenus par chaque protocole individuellement ont peu d'importance. En revanche leur ordre l'est. L'ordre dans lequel on peut classer plusieurs protocoles ne dépend pas de la définition des fonctions d'utilité tant que les conditions (a) et (b) sont réunies, et permet de savoir quel protocole est le plus performant *pour telle ou telle classe de services*.

Les fonctions d'utilité proposées pour les classes de streaming et conversationnelle dépendent d'une multitude de paramètres, par opposition aux classes background et interactive qui, elles, ne dépendent essentiellement que de la quantité totale de données reçues. Pour les premières, l'ensemble de l'historique des allocations de débit est pris en compte dans les valeurs de $v_{s,i}^{(t)}$ et $v_0(s, i, t)$. Ces utilités marginales sont croissantes par rapport au débit, et décroissantes par rapport au délai d_i . De plus, la réduction relative de l'utilité due à une dégradation du service est plus importante pour la classe conversationnelle que pour la classe de streaming. Ces fonctions d'utilité nous permettent d'obtenir un classement sans ambiguïté des protocoles selon les préférences des utilisateurs. Ceci, combiné avec le résultat de la théorie des fonctions d'utilité affirmant que toute combinaison linéaire d'utilités conserve l'ordre assure que l'ordre des valeurs de bénéfice commun est sans ambiguïté, même si les simulations évaluées font intervenir des services appartenant à des classes différentes.

Notez que pour les classes de streaming et conversationnelle, les utilités marginales maximales sont bornées strictement par $2 \cdot v_0$. Un flux au débit minimum r_0 reçoit une utilité de v_0 , alors que toute allocation de débit supplémentaire (au delà de r_0) apporte strictement moins de v_0 . Ainsi, le bénéfice commun atteint sur une durée d'un slot si un utilisateur reçoit le débit minimal de r_0 est plus élevée que si cette bande passante additionnelle était allouée à un ensemble d'utilisateurs déjà servis avec plus de r_0 , ce qui est en accord avec l'objectif d'une métrique mesurant aussi l'équité d'une allocation.

La valeur du bénéfice commun selon les types de trafic et les conditions de la simulation donne une indication sur la sensibilité des applications au comportement d'un protocole. Les conséquences de l'ajout de nouveaux flux, ou de modifications de la topologie sont reflétées par le bénéfice commun. En particulier, quand la charge du réseau devient importante, il arrive un moment où le réseau ne peut plus assurer un débit minimal de r_0 à l'ensemble des flux, ce qui est reflété par une réduction importante du bénéfice commun. La méthodologie proposée ici permet donc d'avoir une idée plus exacte de la qualité de service ressentie par les utilisateurs, à la différence des métriques *en moyenne* ou des indicateurs d'équité comme l'indice de Jain.

7.5 Intérêt de notre méthode d'évaluation

Nous démontrons l'utilité de notre méthode d'évaluation en comparant les performances du protocole *Profiterole* par rapport au standard IEEE 802.11. Il peut cependant être utilisé pour évaluer n'importe quel protocole de répartition de bande passante à tous les niveaux de la pile OSI. Les logiciels utilisés pour cette comparaison sont le simulateur ns2 version 2.33, l'implémentation sous ns2 du protocole *Profiterole* et quelques scripts Perl.

Chaque simulation a une durée de T secondes et se limite à un terrain carré de taille $S \times S$. Pour les deux protocoles étudiés, chaque ensemble de simulations comprend les mêmes ensembles de flux, dont la source et la destination sont choisis aléatoirement. La première seconde de transmission de chaque flux est choisie aléatoirement selon une distribution uniforme dans l'intervalle $[1, T]$. Ainsi, nous pouvons observer l'effet de l'augmentation de la charge au cours de chaque simulation. Les services de classe streaming sont simulés par des flux UDP/CBR, alors que les flux de données sont des flux FTP utilisant TCP comme protocole de transport. Les paramètres utilisés dans ns2 sont donnés en table 7.1, et les paramètres des fonctions d'utilité sont les suivants : $v_0 = 10$, $\alpha_s = 0.97$, $\beta = 0.1$, $\Delta V_s = v_0$, $r_0 = 128kb/s$, $C = 5Mbps$.

TAB. 7.1 – Paramètres de simulation.

Version de NS	2.33
Débit physique	11 Mbps
Débit faisable	5 Mbps
Portée de communication	200 m
Distance de détection de porteuse	397 m
Seuil de capture	10 dB
Modèle de propagation radio	Two Ray Ground
Protocole de routage	AODV
Taille des paquets	1000 octets
RTS/CTS	désactivés

7.5.1 Simulations avec 5, 10 et 15 flux video UDP

Dans la première série de simulations, 10 flux video transmettent à un débit de $384kb/s$ sur un réseau de 30 nœuds déployés sur un terrain de taille $500m \times 500m$, et chaque simulation a une durée de $T = 100sec$. Les valeurs de bénéfice commun calculées sur une moyenne de 10 simulations, ainsi que les intervalles de confiance sont représentés dans la figure 7.2. Les variations de placement des nœuds ainsi que du nombre de flux multi-sauts présents dans chaque expérience expliquent ces larges intervalles de confiance.

TAB. 7.2 – Bénéfice commun moyen pour 10 flux video.

Type d'utilisateur	<i>Bénéfice commun moyen</i>	
	IEEE 802.11	Profiterole
Linéairement adaptatif :	14788	15101
	[10843,19559]	[11757,20300]
Adaptatif par seuil :	14086	14458
	[10344,18574]	[11375,19366]
Fortement adaptatif :	23041	23252
	[16829,30697]	[16987,31598]

Le protocole Profiterole est meilleur que IEEE 802.11 en termes de bénéfice commun. De plus, l'ordre entre les deux protocoles reste le même pour chacune des exécutions de cet ensemble de simulations. En effet, le protocole Profiterole assure une répartition plus équitable de la bande passante. En termes d'utilisation du canal, IEEE 802.11 est légèrement meilleur que Profiterole (17024 paquets par simulation, pour 15998 pour Profiterole). En revanche, les débits alloués avec IEEE 802.11 accusent une variance importante dans les débits alloués aux flux en compétition, et il arrive régulièrement qu'un flux n'atteigne pas la contrainte d'un débit supérieur à r_0 . La figure 7.4 montre l'évolution au cours du temps des débits alloués à chacun des 10 flux pour une simulation. Ce dernier argument, combiné avec les mauvaises performances de IEEE 802.11 en termes d'équité ont pour conséquence une valeur de bénéfice commun plus faible. On constate d'autre part que la variance de IEEE 802.11 croît avec le nombre de flux en compétition. Cette variance est visible à la fois au niveau des allocations individuelles (figure 7.4) qu'au niveau de l'utilisation totale du canal (figure 7.5).

La valeur moyenne de l'indice de Jain pour IEEE 802.11 est de 0.83, et atteint 0.88 pour Profiterole. On observe souvent que IEEE 802.11 est plus performant que Profiterole en termes d'utilisation du médium, mais moins bien en termes d'équité, mais aucune de ces deux métriques ne suffit pour déterminer quel protocole est le plus performant.

Il est intéressant de noter que les valeurs de bénéfice commun de la table 7.2 restent dans le même ordre pour tous les types d'utilisateurs. Comme on pouvait s'y attendre, la valeur du bénéfice commun pour des utilisateurs *adaptatifs par seuils* est plus faible que celle des utilisateurs *linéairement adaptatifs*, puisque la première est une discrétisation de la seconde. De plus, sachant que l'utilité des utilisateurs de type *fortement adaptatifs* augmente rapidement quand le débit alloué dépasse r_0 , c'est le type d'utilisateur pour lequel le bénéfice commun est la plus importante.

Avant de continuer avec d'autres résultats sur la performance de ces protocoles, nous donnons quelques graphiques des débits alloués à une session video avec Profiterole, ainsi que des utilités marginales correspondantes, en supposant que l'utilisateur est du type fortement adaptatif (on observe un comportement

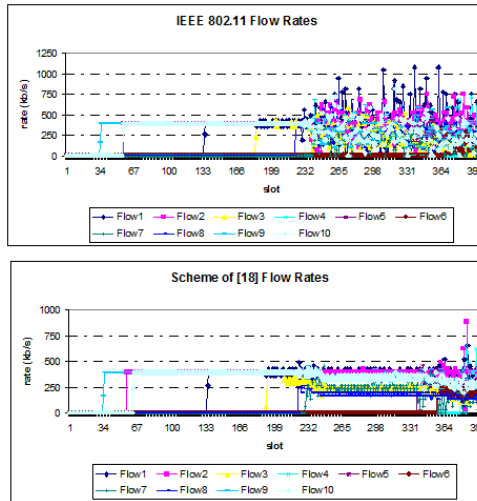


FIG. 7.4 – Evolution des débits au cours du temps pour un réseau de 30 nœuds et 10 flux video.

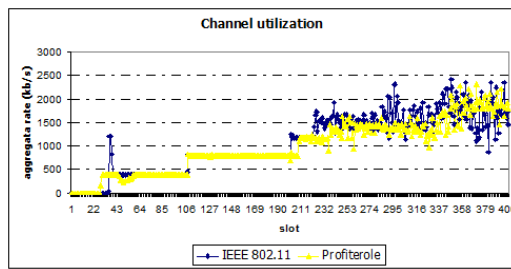


FIG. 7.5 – Utilisation du canal pour un réseau de 30 nœuds et 10 flux video.

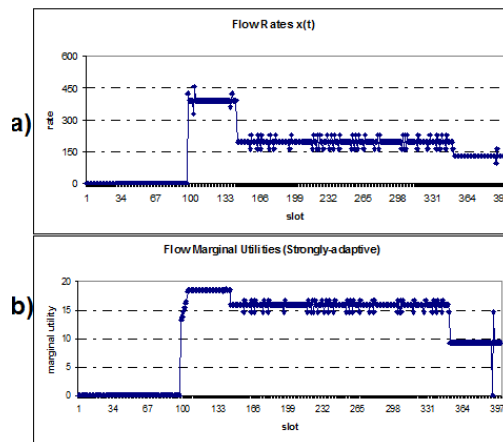
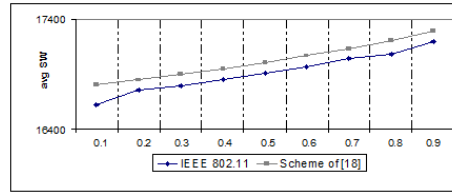


FIG. 7.6 – Débits et utilités marginales d'un flux.

FIG. 7.7 – Bénéfice commun moyen pour différentes valeurs de α_s .

similaire pour tous les flux et les types d'utilisateurs). Ce flux est actif entre la 24^e et la 100^e secondes. Comme les débits atteints sont généralement supérieurs au débit minimal acceptable de $128kb/s$, les valeurs d'utilité marginales (figure 7.6(b)) suivent les tendances des allocations de débit (figure 7.6(a)). Comme de nouveaux flux arrivent progressivement, les débits équitables baissent brusquement, ce qui explique cette forme d'escalier.

Nous avons aussi observé les résultats pour diverses valeurs de α_s qui montrent que l'ordre reste inchangé pour toutes les valeurs de α_s . Les valeurs du bénéfice commun moyen sur l'ensemble des 10 simulations et pour les 3 types d'utilisateurs est représenté en figure 7.7.

Quand la charge réseau est plus faible, les deux protocoles ont des performances similaires. Le tableau 7.3 donne les valeurs moyennes du bénéfice commun sur 10 simulations, avec seulement 5 flux vidéos, et sans changer les autres paramètres. Une fois de plus, IEEE 802.11 est (légèrement) plus efficace en termes d'utilisation du réseau (10679 paquets reçus, à comparer aux 10374 pour Profiterole), mais souffre d'une plus grande variance des allocations. Le gain de performance de Profiterole est moins flagrant ici que dans l'expérience avec 10 flux car la compétition entre flux y est moins féroce.

TAB. 7.3 – Bénéfice commun moyen pour 5 flux video

User Type	<i>Bénéfice commun moyen</i>	
	IEEE 802.11	Profiterole
Linearly-adaptive :	9054 [7542, 12483]	9199 [7446, 14628]
Discretely-adaptive :	8592 [7160, 11845]	8756 [7124, 13624]
Strongly-adaptive :	14231 [11838, 19639]	14410 [11540, 22330]

Dans la série de simulations qui suit, 15 flux video transmettent à des débits de $384kbps$ sur un réseau de 30 nœuds déployés aléatoirement sur un terrain de taille $500m \times 500m$. Chaque simulation dure $T = 100sec$. Les valeurs moyennes du bénéfice commun sur un total de 10 simulations, ainsi que les intervalles de confiance sont résumés en table 7.4.

TAB. 7.4 – Bénéfice commun moyen pour 15 flux video.

Type d'utilisateur	<i>Bénéfice commun moyen</i>	
	IEEE 802.11	Profiterole
Linearly-adaptive	19646 [15944, 26326]	22549 [17961, 28173]
Discretely-adaptive	18819 [15291, 25275]	21751 [17294, 27272]
Strongly-adaptive	30238 [24539, 40224]	33894 [27012, 41000]

Une fois de plus Profiterole est meilleur que IEEE 802.11, avec la différence que les valeurs de bénéfice commun sont plus élevées que celles des simulations à 5 et 10 flux. De plus, le nombre moyen de paquets reçus avec Profiterole est cette fois-ci plus élevé qu'avec IEEE 802.11. Ainsi, utiliser un protocole de

partage équitable de bande passante quand plusieurs flux sont en compétition peut être avantageux non seulement en termes d'équité, mais aussi en termes d'utilisation du canal.

Il est intéressant de comparer les valeurs de bénéfice commun atteintes pour les simulations avec 10 flux video UDP. Il est clair que la valeur du bénéfice commun pour les deux protocoles a augmenté. Cela nous indique que le réseau n'est pas saturé, car le bénéfice commun augmente quand on ajoute de nouveaux flux. Dans le cas contraire, on aurait observé des valeurs de bénéfice commun très faibles.

7.5.2 Simulations avec une combinaison de flux UDP/CBR et TCP/FTP

Enfin, les simulations indiquent que Profiterole est meilleur que 802.11 quand le réseau est utilisé par une combinaison de flux TCP et de flux UDP. Dans la série de simulations qui suit, 10 flux video transmettant à des débits de 384kbps et 5 flux FTP/TCP sont en compétition dans un réseau de 30 nœuds, déployés une fois de plus aléatoirement sur un terrain de taille $500\text{m} \times 500\text{m}$. Un nouveau paramètre apparaît dans cette simulation : pour les flux *FTP* on choisit d'attribuer une utilité de 0.01 par *kbps* et par seconde. Les résultats de ces simulations sont en table 7.5.

TAB. 7.5 – Bénéfice commun pour 10 flux video/UDP et 5 flux FTP/TCP.

User Type	<i>Bénéfice commun moyen</i>	
	IEEE 802.11	Profiterole
Linearly-adaptive	12392	13164
	[10443,15069]	[10053,15993]
Discretely-adaptive	11860	12758
	[10016,14450]	[9743,15407]
Strongly-adaptive	18931	19307
	[15830,22823]	[14656,23396]

Le trafic est une combinaison de flux TCP et UDP et les applications ont des propriétés et une sensibilité à la qualité de service différentes. Il n'est donc pas possible d'appliquer une métrique comme l'indice de Jain dans un tel contexte. En revanche, le bénéfice commun permet ici de comparer les performances des deux protocoles.

Dans l'ensemble, cette méthode d'évaluation permet d'observer les propriétés intrinsèques des protocoles étudiés. Les protocoles équitables tels que Profiterole allouent les débits de manière stable au cours du temps ; cette propriété est bénéfique en particulier pour les flux sensibles à la QoS, ainsi que pour les flux multi-sauts. Cette propriété s'accompagne d'une baisse relativement faible de l'utilisation du canal quand le réseau est modérément congestionné, baisse que l'on n'observe pas toujours quand la compétition est forte.

7.6 Conclusions

Dans ce chapitre, nous avons présenté un cadre d'évaluation reposant sur des fonctions d'utilité à mémoire. Ces fonctions quantifient la satisfaction à long terme d'un utilisateur de réseau sans fil selon la manière dont lui est allouée la bande passante, de sorte à tenir compte de la mobilité et donc de l'instabilité des réseaux sans fil. Nous pensons que cette approche est nouvelle et qu'elle peut servir d'outil d'évaluation de performance pour les protocoles équitables des réseaux sans fil. Comme l'a montré la comparaison du protocole Profiterole avec le standard IEEE 802.11, cette approche permet à la différence des métriques habituelles des réseaux sans fil de quantifier l'impact des variations à court terme des allocations de bande passante.

Conclusions

Dans ce manuscrit, j'ai présenté les difficultés que soulève la norme 802.11 dans les réseaux ad hoc multi-sauts. Les interactions dans ces réseaux sont de deux sortes : la contention, qui est une conséquence du principe CSMA, et les interférences, qui sont à l'origine des collisions. Les inégalités de partage de bande passante proviennent de ces deux types d'interactions, et de manière générale, du fait qu'il est fréquent dans ces réseaux qu'une ou plusieurs stations interfèrent - ou soient en contention - avec une autre station sans qu'aucune information locale ne permette de le détecter. Ainsi, ces stations n'ont aucun moyen de savoir qu'il est nécessaire de réduire leur débit pour un partage plus équitable. C'est pourquoi les solutions utilisant des informations locales peinent à résoudre ce type de difficultés.

La régulation de bande passante à un niveau plus haut que le niveau MAC est une manière de répondre à ce problème, par exemple en colportant au niveau routage les informations qui vont permettre de détecter ce type d'interactions asymétriques. Cette méthode suppose d'inférer les interactions dans le réseau à partir d'informations disponibles au niveau routage comme les informations de topologie. J'ai présenté différents modèles d'interactions à base de graphes de conflit, plus ou moins simples, et modélisant plus ou moins bien la propagation radio. Partant de la constatation que la propagation radio est difficile à modéliser, et qu'il vaut mieux ne pas trop se fier à des informations telles que la distance ou les niveaux de SINR, j'ai proposé un modèle simple d'utilisation, qui malgré sa simplicité, n'est pas plus imprécis que les modèles actuels proposés dans la littérature. Notamment, l'utilisation de la métrique du nombre de sauts a l'avantage de s'affranchir au maximum d'un modèle de propagation radio.

Enfin, j'ai présenté une implémentation protocolaire distribuée, appelée *Profiterole*, permettant de réguler dans son ensemble chacun des débits des flux à leur valeur optimale en termes d'équité et d'efficacité. A l'aide de simulations, nous avons comparé cette approche à la norme 802.11 et au protocole DiffQ - une approche de niveau MAC - et constaté que *Profiterole* améliore nettement l'équité du partage de bande passante. Nous avons vu que l'équité du partage de la bande passante est un problème crucial, à la fois du point de vue de la satisfaction des utilisateurs du réseau, mais aussi pour garantir une stabilité au réseau et permettre un routage plus fiable.

Mais est-il pertinent de ne s'intéresser qu'au partage de la bande passante? En effet, *Profiterole* ne permet que de partager la capacité entre plusieurs flux *Best Effort*. Qu'en est-il des flux nécessitant une certaine qualité de service, des flux plus sensibles au délai et à la gigue qu'à la quantité de bande passante? Tout d'abord, nous avons vu lors du dernier chapitre que *Profiterole* répondait dans une certaine mesure à ce problème puisque la régulation de débit, en limitant la bande passante, limite aussi la congestion et les délais. D'autre part, une amélioration notable de *Profiterole* consisterait à prendre en compte l'existence de flux nécessitant une bande passante fixe dans les équations de sorte à répondre au problème : étant donné qu'un certain nombre de flux *QoS* traversent le réseau, comment allouer le reste de la bande passante à d'autres flux *Best Effort* en gênant le moins possible les flux *QoS*? Ce problème semble soluble avec *Profiterole*, quoique les imprécisions du modèle de partage ne permettent pas de donner des garanties absolues aux flux *QoS*.

Aujourd'hui, l'obstacle principal à une implémentation réelle de *Profiterole* est la lenteur de convergence de l'algorithme. A l'échelle de la vie d'un flux, une durée de quelques secondes est en effet significative. Ce défaut, même s'il est moins prononcé que pour d'autres solutions de ce type utilisant des modèles plus complexes, est particulièrement gênant pour des réseaux très mobiles. Ceci, combiné avec le fait qu'on s'intéresse à la régulation de flux élastiques dont le critère principal est la bande passante

fait que ce type de protocole est destiné à des réseaux de type mesh plutôt qu'à des réseaux de capteurs, ou des réseaux de véhicules. Le problème de la faible convergence peut cependant devenir moins critique avec le développement de technologies 802.11 plus rapides, où l'augmentation de la capacité permet l'utilisation de paquets Hello plus fréquents et donc une meilleure convergence.

Pour conclure sur ce travail, j'ajouterai qu'il permet dans une certaine mesure de répondre à une question fondamentale des réseaux ad hoc. A quel niveau du modèle OSI doivent se trouver les mécanismes pour résoudre les problèmes d'inégalité de partage dans les réseaux ad hoc ?

Nous avons identifié la nécessité d'échanger des informations au delà de la portée de communication et donc l'utilité de placer cette régulation au niveau 3, i.e. au niveau routage. Mais les modèles d'interaction sont nécessairement imprécis s'ils n'utilisent que les information topologiques. Il arrive donc régulièrement que *Profiterole* sous-estime le débit auquel un flux peut émettre, de la même manière que les RTS/CTS ne réservent le canal que pour les stations situées à portée de communication du récepteur, alors que les protocoles de couche 2 comme DiffQ ou MadMAC ne dépendent pas d'un modèle de propagation radio.

Inversement, la couche 2 ne permet pas de faire une régulation de flux, mais plutôt une régulation par station. Ce manque de connaissance globale amène aux résultats que l'on a constatés, où DiffQ donne une priorité trop grande aux flux multi-sauts. Il me semble aujourd'hui que la résolution des difficultés dans les réseaux ad hoc passe soit par une étroite collaboration entre les couches 2 et 3, ou pourquoi pas, puisque le routage, et la qualité des liens sont si intimement liés, par une refonte du modèle OSI, établi à l'origine pour les réseaux filaires ?

Annexe A

Annexes

A.1 Preuve du théorème de convergence avec erreur

Théorème énoncé en section 5.2.3 :

Théorème 14. *On suppose qu'il existe un entier β tel que à l'instant t , toutes les entités exécutant l'algorithme ont des informations sur leur voisinage ne datant pas de plus de β pas de l'algorithme. Alors il existe ϵ tel que pour tout pas $\alpha < \epsilon$ l'algorithme asynchrone converge vers le minimum de \mathcal{L} .*

Démonstration. L'idée de la preuve est inspirée des preuves de convergence avec erreur du livre de Bertsekas [9]. Elle consiste à montrer que malgré l'erreur, la solution $\Lambda^{(t)}$ s'améliore à chaque pas, i.e. si l'on pose :

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \alpha \nabla \mathcal{L} + e^{(t)}$$

où $e^{(t)}$ est l'erreur au pas t , alors :

$$\mathcal{L}(\Lambda^{(t+1)}) < \mathcal{L}(\Lambda^{(t)})$$

Comme on peut montrer facilement que tout point stationnaire de la suite $\Lambda^{(t)}$ est un minimum local de \mathcal{L} , cette dernière condition suffit à prouver que l'algorithme converge.

Supposons donc que la station i ait des informations un peu obsolètes sur les tarifs de son voisinage, mais ne datant pas plus de β pas de l'algorithme. Donc l'erreur sur Λ , $\Delta\Lambda$ peut être majorée par :

$$\Delta\Lambda = O(\|\alpha\beta\nabla\mathcal{L}\|)$$

De la même manière, l'erreur sur π_f :

$$\Delta\pi_f = O(\|\alpha\beta\nabla\mathcal{L}\|)$$

D'autre part, au voisinage de l'optimal, $\pi_f < \frac{1}{C}$, puisque π_f est l'inverse du débit optimal du flux f , et que celui-ci ne peut excéder la capacité C . Aussi, on peut donc écrire d'après l'équation, rappelée ici :

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = C - \sum_f \frac{1}{\pi_f} r_{if}$$

que l'erreur sur $\frac{\partial \mathcal{L}}{\partial \lambda_i}$, $\Delta d\Lambda$ est elle aussi de l'ordre de :

$$\Delta d\Lambda = O(\|\alpha\beta\nabla\mathcal{L}\|)$$

Ainsi, si α est choisi suffisamment petit, on peut vérifier pour tout t :

$$\|e^{(t)}\| < \|\nabla\mathcal{L}(\Lambda^{(t)})\|$$

condition qui assure que l'algorithme reste dans une direction de descente (résultat prouvé dans [9], page 34). CQFD. \square

A.2 Implémentation du token bucket dans Profiterole

Soit t l'instant où un paquet arrive à la couche AODV en provenance des couches supérieures. On utilise un timestamp T qui représente la date à laquelle le prochain paquet devra être envoyé. Les valeurs θ^+ et θ^- sont des seuils de tolérance autour de cette date T . On suppose que le temps d'occupation du medium pour un paquet est connu, et vaut t_p . Dans l'implémentation actuelle du protocole, ce temps est évalué en fonction de la taille du paquet reçu et du protocole de transport utilisé (les paquets TCP prennent plus de temps d'occupation du medium dans la mesure où chaque paquet doit être acquitté).

A la réception d'un paquet, on distingue trois cas :

- $t > T - \theta^-$: alors $T \rightarrow t + \theta^- + t_p$.
- $T - \theta^+ < t < T - \theta^-$: alors $T \rightarrow T + t_p$.
- $t < T - \theta^+$: le paquet est jeté et T ne change pas.

Table des figures

1.1	Carte de probabilité de réception théorique face à celle d'une antenne réelle	5
1.2	Couches physiques de 802.11	7
1.3	Temps d'attente dans 802.11b	9
1.4	Scénario des stations cachées	10
1.5	Scénario de la station exposée	11
2.1	Durée d'une communication atomique dans 802.11b	14
2.2	Evolution du débit agrégé selon le nombre d'émetteurs en contention	15
2.3	Deux scénarios problématiques des réseaux multi-sauts	17
4.1	Graphe de conflit en étoile	30
4.2	Un graphe de conflit imparfait : le trou de longueur 5	31
4.3	Choix de k : correspondance entre densité et nombre de stations	35
4.4	Correspondance entre distance et nombre de sauts (1)	35
4.5	Correspondance entre distance et nombre de sauts (2)	36
4.6	Compromis entre efficacité et équité dans la famille de fonctions objectifs de l'équation 4.6	39
5.1	Un exemple de graphe de conflit	45
5.2	Implémentation protocolaire de l'algorithme du gradient	50
6.1	Topologie pour le flux à quatre sauts	57
6.2	Evaluation de <i>Profiterole</i> sur un flux à 4 sauts - flux UDP.	58
6.3	Evaluation de <i>Profiterole</i> sur un flux à 4 sauts - flux TCP.	58
6.4	Flux TCP à 20 stations (19 sauts).	59
6.5	Stations cachées asymétriques.	60
6.6	Résultats de l'évaluation pour les stations cachées asymétriques.	60
6.7	Temps d'attente dans quelques variantes de 802.11	61
6.8	Tarifs et prix lors d'une exécution de la simulation 4 flux - 25 stations.	63
6.9	Fonction cumulative de distribution des débits des flux pour un réseau de 25 stations	63
6.10	Scénario étudiés.	65
6.11	Evolution du débit en fonction du temps dans le scénario des trois paires.	65
6.12	Evolution du débit en fonction du temps dans le scénario de la chaîne de communication.	66
6.13	Evolution du débit en fonction du temps sur un scénario aléatoire (30 noeuds, 5 flux, udp).	67
6.14	Evolution du débit en fonction du temps sur un scénario aléatoire (30 noeuds, 5 flux, udp)	67
7.1	Débit moyen selon les types d'allocation dans le scénario des trois paires	69
7.2	Fonctions d'utilité typiques. Source : [46].	71
7.3	Structure de notre méthode d'évaluation à base de fonctions d'utilité.	72
7.4	Evolution des débits au cours du temps pour un réseau de 30 noeuds et 10 flux video.	77
7.5	Utilisation du canal pour un réseau de 30 noeuds et 10 flux video.	77
7.6	Débits et utilités marginales d'un flux.	77
7.7	Bénéfice commun moyen pour différentes valeurs de α_s	78

Bibliographie

- [1] 3gpp specification 23.107 quality of service (qos) concept and architecture. <http://www.3gpp.org>.
- [2] The ieee 802.11 standards. <http://standards.ieee.org/getieee802/802.11.html>.
- [3] Daniel Aguayo, John Bicket, Sanjit Biswas, and Douglas De Couto. Mit roofnet : Construction of a production quality ad-hoc network, 2003.
- [4] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. Ieee 802.11b ad hoc networks : Performance measurements. *Cluster Computing*, 8(2-3) :135–145, 2005.
- [5] Emilio Ancillotti, Raffaele Bruno, and Marco Conti. Experimentation and performance evaluation of rate adaptation algorithms in wireless mesh networks. In *PE-WASUN*, pages 7–14, 2008.
- [6] Elyes Ben Hamida, Guillaume Chelius, and Jean-Marie Gorce. Impact of the Physical Layer Modeling on the Accuracy and Scalability of Wireless Network Simulation. *Simulation*, 85 :574–588, September 2009.
- [7] Brahim Bensaou, Y. Wang, and C. C. Ko. Fair Medium Access in 802.11 based Wireless Ad-Hoc Networks. In *ACM MobiHoc*, pages 99–106, Boston, Massachusetts, 2000.
- [8] G. Berger-Sabbatel, A. Duda, O. Gaudoin, M. Heusse, and F. Rousseau. Fairness and its impact on delay in 802.11 networks. In *Proc. of the IEEE Globecom*, pages 2697–2973, 2004.
- [9] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, September 1999.
- [10] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW : a media access protocol for wireless LAN's. In *ACM SIGCOMM Computer Communication Review*, pages 212–225, London, United Kingdom, 1994.
- [11] D. R. Boggs, J. C. Mogul, and C. A. Kent. Measured capacity of an Ethernet : myths and reality. In *Proc. SIGCOMM '88 Symposium on Communications Architectures and Protocols*, volume 18, pages 222–34, 1988.
- [12] Heinz Breu and David G. Kirkpatrick. Unit disk graph recognition is np-hard. *Computational Geometry. Theory and Applications*, 9, 1993.
- [13] Jinhai Cai, Dominic Ee, Binh Pham, and Paul Roe. Sensor network for the monitoring of ecosystem : Bird species recognition. *Third International Conference on Intelligent Sensors, Sensor Networks and Information Processing 2007 (ISSNIP'07)*, 2007.
- [14] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain. Watching television over an ip network. In *Proc. of the 8th ACM SIGCOMM conference on Internet measurement*, pages 71–84, Vouliagmeni, Greece, October 2008.
- [15] P. Chatzimisios, A.C. Boucouvalas, V. Vitsas, A. Vafiadis, A. Oikonomidis, and P. Huang. A simple and effective backoff scheme for the IEEE 802.11 MAC protocol. In *CITSA*, Orlando, Florida, USA, July 2005.
- [16] C. Chaudet. *Autour de la réservation de bande passante dans les réseaux ad hoc*. PhD thesis, INSA de Lyon, INRIA Rhône Alpes, France, September 2004.
- [17] C. Chaudet, G. Chelius, H. Meunier, and D. Simplot-Ryl. Adaptive Probabilistic NAV to Increase Fairness in Ad Hoc 802.11 Mac Layer. In *MedHocNet*, 2005.
- [18] C. Chaudet, D. Dhoutaut, and I. Guérin Lassous. Performance Issues with IEEE 802.11 in Ad Hoc Networking. *IEEE Communication Magazine*, 43(7) :110–116, July 2005.

- [19] C. Chaudet and I. Guérin-Lassous. Influence de l'asymétrie et des interférences sur l'équité de l'accès au médium dans les réseaux 802.11b. In *Algotel 2003*, Banyuls-sur-Mer, France, 2003.
- [20] Claude Chaudet and Isabelle Guérin Lassous. Bruit : Bandwidth reservation under interferences influence. In *Proc. of the European Wireless (EW02)*, pages 466–472, 2002.
- [21] Qi Chen, Felix Schmidt-Eisenlohr, Daniel Jiang, Marc Torrent-Moreno, Luca Delgrossi, and Hannes Hartenstein. Overhaul of iee 802.11 modeling and simulation in ns-2. In *MSWiM '07 : Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 159–168, New York, NY, USA, 2007. ACM.
- [22] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3) :165–177, December 1990.
- [23] C. Courcoubetis and R. Weber. *Pricing Communication Networks : Economics, Technology and Modelling*. Wiley and Sons, 2003.
- [24] Dei 802.11 multi-rate, a new 802.11 implementation for ns-2. <http://www.dei.unipd.it/wdyn/?IDsezione=5090>.
- [25] Javier Del Prado Pavon and Sunghyun Choi. Link adaptation strategy for iee 802.11 wlan via received signal strength measurement. In *IEEE ICC'03*, 2003.
- [26] D. Dhoutaut and I. Guérin-Lassous. Impact of heavy traffic beyond communication range in multi-hops ad hoc networks. In *Third International Network Conference*, Plymouth, Angleterre, 2002.
- [27] D Dhoutaut and Isabelle Guérin-Lassous. Experiments with 802.11b in ad hoc configurations. In *14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1618–1622, Beijing, China, September 2003. IEEE Press.
- [28] E. Dramitinos, I. Guérin Lassous, and R. Vannier. A Performance Evaluation Framework for Fair Solutions in Ad Hoc Networks. In *12-th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Tenerife, Canary Islands, Spain, October 2009.
- [29] M. Dramitinos, G. D. Stamoulis, and C. Courcoubetis. Auction-based resource reservation in 2.5/3g networks. *Kluwer/ACM Mobile Networks and Applications Special Issue on Mobile and Pervasive Commerce*, 9(6) :557–566, December 2004.
- [30] M. Dramitinos, G. D. Stamoulis, and C. Courcoubetis. Auction-based resource allocation in umts high speed downlink packet access. In *Proc. of the First EuroNGI Conference Conference on Traffic Engineering*, pages 434–441, Rome, Italy, April 2005.
- [31] Matthew Gast. *802.11 Wireless Networks : The Definitive Guide, Second Edition (Definitive Guide)*. O'Reilly Media, Inc., April 2005.
- [32] Stefanie Gerke and Colin McDiarmid. Graph imperfection. *Journal of Combinatorial Theory Series B*, 83(1) :58–78, 2001.
- [33] I. Guérin Lassous, T. Razafindralambo, and R. Vannier. Evaluation d'un protocole de régulation de débit dans les réseaux sans fil multisautes. In *Conférence Francophone sur l'Ingénierie des Protocoles*, Strasbourg, France, oct 2009.
- [34] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 34(5) :910–917, 2000.
- [35] Rajarshi Gupta, John Musacchio, and Jean Walrand. Sufficient rate constraints for qos flows in ad-hoc networks. *Ad Hoc Networks*, 5(4) :429–443, May 2007.
- [36] Elyes Ben Hamida, Guillaume Chelius, and Jean-Marie Gorce. Scalable versus accurate physical layer modeling in wireless network simulations. In *PADS '08 : Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation*, pages 127–134, Washington, DC, USA, 2008. IEEE Computer Society.
- [37] Ahmad Al Hanbali and Eitan Altman. A survey of tcp over ad hoc networks. *IEEE Communications Surveys and Tutorials*, 7 :22–36, 2005.

- [38] J. He and H. K. Pung. Fairness of Medium Access Control Protocols for Multi-hop Ad Hoc Wireless Networks. *Computer Networks*, 48(6) :867–890, 2005.
- [39] Karel Heurtefeux and Fabrice Valois. Localisation distribuée pour routage en environnement bruité dans les réseaux de capteurs. In *CFIP'2009*, Strasbourg France, oct 2009.
- [40] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, volume 2, pages 836–843 vol.2, 2003.
- [41] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks : shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.
- [42] H. Knoche and J. D. McCarthy. Design requirements for mobile tv. In *Proc. of the 7th international conference on Human computer interaction with mobile devices and services*, pages 69–76, Salzburg, Austria, 2005.
- [43] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *MSWiM '04 : Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 78–82, New York, NY, USA, 2004. ACM Press.
- [44] S. Kumar, V. S. Raghavan, and J. Deng. Medium Access Control protocols for ad hoc wireless networks : A survey. *Ad Hoc Networks*, 4 :326–358, 2006.
- [45] K. Lee. Adaptive network support for mobile multimedia. In *Proceedings of the 1st annual international conference on Mobile computing and networking*, pages 62–74, 1995.
- [46] R. F. Liao and A. T. Campbell. A utility-based approach for quantitative adaptation in wireless packet networks. *Kluwer Wireless Networks*, 7 :541–557, 2001.
- [47] Xiaojun Lin and Ness B. Shroff. Joint rate control and scheduling in multihop wireless networks. In *in Proceedings of IEEE Conference on Decision and Control*, pages 1484–1489, 2004.
- [48] H. Luo, S. Lu, and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. In *MOBICOM*, pages 76–86, Boston, Massachusetts, United States, 2000.
- [49] P. Maillé and B. Tuffin. An auction-based pricing scheme for bandwidth sharing with history-dependent utility functions. In *Proc. of the First International Workshop on Incentive Based Computing (IBC'05)*, pages 30–49, Compiègne, France, September 2005.
- [50] Pradeepkumar Mani and David W. Petr. Clique number vs. chromatic number in wireless interference graphs : Simulation results. Technical Report ITTC-FY2007-TR41420-01, Information Telecommunication and Technology Center, University of Kansas, Lawrence, KS, 2007.
- [51] M.V. Marathe, H. Breu, H. B. Hunt Iii, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25 :59–68, 1995.
- [52] D. Miras, R. J. Jacobs, and V. Hardman. Content-aware quality adaptation for ip sessions with multiple streams. In *Proc. of the 8th International Workshop on Interactive Distributed Multimedia Systems*, pages 168–180, London, UK, 2001. Springer-Verlag.
- [53] Jeonghoon Mo and Jean C. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Netw.*, 8(5) :556–567, 2000.
- [54] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *MOBICOM*, pages 87–98, Boston, Massachusetts, United States, 2000.
- [55] Mohammad Nassiri, Martin Heusse, and Andrzej Duda. Forced transmissions for coping with the effect of blocked stations in 802.11 wireless networks. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 0 :1–9, 2007.
- [56] The opnet modeler. http://www.opnet.com/solutions/network_rd/modeler.html.
- [57] Qualnet scalable network technologies. <http://www.qualnet.com/>.
- [58] William R. Hawe Rajendra K. Jain, Dah-Ming W. Chiu. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems.

- [59] T. Razafindralambo and I. Gu erin Lassous. Increasing Fairness and Efficiency using the MadMac Protocol in Ad Hoc Networks. *Ad Hoc Networks Journal, Elsevier Ed.*, 6(3) :408–423, May 2008.
- [60] T. Razafindralambo and I. Gu erin Lassous. SBA : a Simple Backoff Algorithm for Wireless Ad Hoc Networks. In *IFIP Networking*, Aachen, Germany, May 2009.
- [61] Tahiry Razafindralambo and Isabelle Gu erin-Lassous. Increasing fairness and efficiency using the madmac protocol in ad hoc networks. In *IFIP Networking*, pages 403–414, May 2006.
- [62] J. Sachs, M. Prytz, and J. Gebert. Multi-access management in heterogeneous networks. *Wireless Personal Communications*, 48(1) :7–32, January 2009.
- [63] S. Shenker. Fundamental design issues for the future internet. *IEEE Selected Areas in Communications*, 13(7) :1176–1188, September 1995.
- [64] Leslie G. Valiant. Universality considerations in vlsi circuits. *IEEE Trans. Computers*, 30(2) :135–140, 1981.
- [65] R. Vannier and I. Gu erin Lassous. Partage  equitable de la bande passante dans les r eseaux ad hoc. In *CFIP*, Les Arcs, France, March 2008.
- [66] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, second edition, 1947.
- [67] W.E. Walsh, G. Tesauro, J.O. Kephart, and R. Das. Utility functions in autonomic systems. In *Proc. of the International Conference on Autonomic Computing*, pages 70–77, 2004.
- [68] Ajit Warriar, Sangtae Ha, P. Wason, Injong Rhee, and J. H. Kim. Diffq : Differential backlog congestion control for wireless multi-hop networks. In *SECON*, pages 585–587. IEEE, 2008.
- [69] D. B. West. *Introduction to Graph Theory (2nd Edition)*. Prentice Hall, 2001.
- [70] G. Wikstrand and J. Sun. Determining utility functions for streaming low bit rate soccer video. In *Proc. of IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA)*, page August, Kauai, Hawaii, USA, 2004.
- [71] Kun-Da Wu and Wanjiun Liao. Flow allocation in multi-hop wireless networks : A cross-layer approach. *Wireless Communications, IEEE Transactions on*, 7(1) :269–276, Jan. 2008.
- [72] Kaixin Xu, M. Gerla, and Sang Bae. How effective is the ieee 802.11 rts/cts handshake in ad hoc networks. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 1, pages 72–76 vol.1, 2002.
- [73] Y. Xue, B. Li, and K. Nahrstedt. Optimal Resource Allocation in Wireless Ad Hoc Networks : A Price-based Approach. *IEEE Transactions on Mobile Computing*, 5(4) :347– 364, April 2006.
- [74] X. Yang and N. Vaidya. On physical carrier sensing in wireless ad hoc networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2525–2535 vol. 4, 2005.
- [75] Yong Yang, Jennifer C. Hou, and Lu-Chuan Kung. Modeling the effect of transmit power and physical carrier sense in multi-hop wireless networks. In *INFOCOM*, pages 2331–2335, 2007.