

# Etude de l'équité dans les réseaux ad-hoc

Mathias Péron

*Rapport de stage MIM1 - 2003*

**Résumé** Ce rapport souhaite donner une description globale et précise du problème de l'équité au niveau de la couche réseau MAC et dresser l'éventail des algorithmes jusqu'ici proposés.

Après une courte présentation des réseaux ad-hoc, le lecteur trouvera la description constructive du protocole MAC standard de 802.11 suivant les problèmes qui ont été rencontrés. La seconde partie soulève des questions d'ordre général sur l'équité et explore l'ensemble des algorithmes connus. Suit une description formelle du phénomène d'inégalité. Les deux sections suivantes traitent d'inégalité due à des raisons protocolaires ou intrinsèques aux réseaux ad-hoc. Enfin les derniers paragraphes sont consacrés à l'algorithme que nous proposons et à décrire ce que de futurs travaux devraient aborder.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>le protocole de l'IEEE 802.11</b>	<b>3</b>
2.1	Vue d'ensemble de DFWMAC . . . . .	4
2.2	Diffusion radio et conséquences . . . . .	5
2.3	Des interférences . . . . .	7
2.4	De la mobilité . . . . .	9
<b>3</b>	<b>Ce que devrait être l'équité</b>	<b>9</b>
<b>4</b>	<b>Ce qu'est l'inégalité</b>	<b>10</b>
4.1	Un scénario simple . . . . .	10
4.2	Approche théorique de l'équité . . . . .	11
<b>5</b>	<b>Une inégalité protocolaire</b>	<b>13</b>
<b>6</b>	<b>Une inégalité intrinsèque</b>	<b>16</b>
<b>7</b>	<b>Une solution</b>	<b>17</b>
7.1	Un algorithme de résolution de fenêtre de contention . . . . .	18
7.2	Résultats . . . . .	20
<b>8</b>	<b>Conclusion</b>	<b>22</b>
<b>A</b>	<b>Curiosités</b>	<b>24</b>
<b>B</b>	<b>NetworkSimulator 2</b>	<b>24</b>
<b>C</b>	<b>Algorithme</b>	<b>25</b>
<b>D</b>	<b>Equipe d'accueil et remerciements</b>	<b>27</b>

# 1 Introduction

Les réseaux ad-hoc sont des réseaux locaux utilisant le médium radio. Ils sont constitués d'entités sans fil et mobiles ne détenant aucun rôle préalable. Aucune station de base n'entre en jeu.

Ainsi, de par leur mobilité, leur déploiement rapide et le fait qu'ils ne nécessitent aucune infrastructure fixe, les réseaux ad-hoc trouvent de nombreuses applications tant dans le civil, par exemple lors de catastrophes naturelles pour déployer les secours (avec des problématiques comme le rattachement à un réseau filaire. On parle de réseaux hybrides\*) que dans le monde militaire où l'on a la conviction de l'enjeu de maîtriser la technologie ad-hoc (avec des applications comme les réseaux de capteurs: sensor-networks\*).

Les problématiques ouvertes sont nombreuses (comme par exemple la phase d'admission d'un nœud, les garanties de sécurité et confidentialité et la qualité de service (QoS)). Pourtant, la plupart des recherches se sont focalisées jusqu'ici sur les problèmes de routage. Elles ont été menées sous l'hypothèse que la gestion du médium, confiée à la couche MAC (Medium Access Control), était sans faille.

Une couche MAC a été définie par l'IEEE dans le standard 802.11. Cependant, des recherches ont montré que son fonctionnement rencontrait de nombreuses défaillances dans le cadre des réseaux ad-hoc. Celle qui nous intéresse est le problème de l'accès équitable au médium de communication pour tous les mobiles du réseau. En effet, ne pas permettre à tous les mobiles d'accéder au réseau peut créer des dysfonctionnements importants.

Nous allons donc débiter notre réflexion par la compréhension du standard 802.11 dans la section 2. Nous approfondirons notre savoir des phénomènes qui mettent à mal l'équité dans les sections 3, 5 et 6. Pour terminer nous décrirons la solution proposée.

## 2 le protocole de l'IEEE 802.11

Le protocole 802.11 est un standard de l'IEEE qui définit une couche physique et une couche MAC pour les réseaux locaux sans fil. Le premier standard a été défini en 1997. Il opère sur la bande de fréquence 2.4 GHz et permet des débits de 2 Mb/s.

Actuellement les fabricants de cartes réseaux sans fil s'en tiennent au standard 802.11b qui travaille dans la même bande de fréquence mais permet un débit de 11 Mb/s. Le protocole MAC de 802.11, d'ailleurs identique à celui de 802.11b, est connu sous le nom de DFWMAC.

## 2.1 Vue d'ensemble de DFWMAC

DFWMAC est un protocole de type CSMA/CA (Carrier Sense Multiple Access) c'est à dire qui opère dans un environnement à médium partagé où il tente de limiter les collisions (Collision Avoidance). En effet, il est impossible pour un mobile qui transmet de *détecter* s'il y a eu ou non collision, puisque la collision éventuelle se situe au niveau du mobile récepteur, et parceque le médium radio ne permet pas à l'émetteur d'écouter le médium pendant qu'il l'utilise. Le protocole ne peut donc qu'essayer d'*éviter* les collisions.

Dans cette optique, le protocole utilise des paquets d'acquiescement – on assure que le paquet de donnée a été transmis avec succès – et un temps aléatoire afin de réguler les envois – on réduit la probabilité de collision. L'algorithme d'envoi d'un paquet de donnée suivi d'un chronogramme explicatif sont donnés ci-dessous.

---

```
Attendre que le médium soit libre
Attente d'un temps fixe DIFS
    pendant lequel le médium doit rester libre
Tirage aléatoire dans [0,CW] (Contention Window) de la valeur BACKOFF
    (fenêtre de contention)

Tant que le médium est libre
    décrémenter BACKOFF

Si BACKOFF est nul
    envoi du paquet DATA
    attente SIFS, correspondant au temps de transmission du paquet
    réception du paquet ACK

Sinon retour au début, avec BACKOFF *restant*
```

---

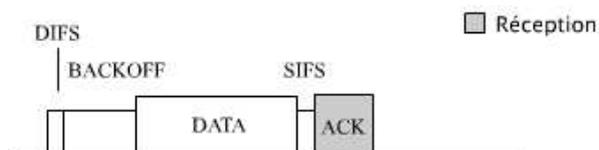


FIG. 1 – Cycle de transmission d'un paquet

L'idée générale est d'instaurer un temps d'attente où le médium doit être libre et statistiquement différent pour chaque mobile. Le tirage aléatoire s'effectue dans une fenêtre appelée fenêtre de contention CW. On assure ainsi une distribution équitable du médium. Un nœud ne peut pas non plus se trouver en situation de famine puisque son temps d'attente BACKOFF diminue à chaque échec de prise du médium jusqu'à la valeur nulle. Enfin la réception de l'acquiescement (ACK) scelle la réussite de la transmission.

## 2.2 Diffusion radio et conséquences

Lorsque l'on transmet sur le médium radio, le signal est diffusé dans un espace que l'on peut assimiler à une sphère. Ci-dessous on a représenté un flux du nœud a au nœud b, avec leurs sphères de transmission/réception (on supposera leurs tailles identiques).

Tout nœud dans la zone (i) tombe sous le coup de la diffusion, broadcast en anglais, radio du nœud a. Il peut donc établir une communication avec a, mais il ne peut recevoir, c'est à dire interpréter, de signaux d'autres nœuds. On parle de nœuds exposés à une transmission.

La seconde zone (ii) délimite la portée de réception du nœud b. Ainsi s'il se trouve des nœud émetteurs dans cet espace, leur signal interférera en b avec celui produit par a et il y a collision.

Enfin notons que pendant l'émission de l'ACK, ces deux zones s'inversent.

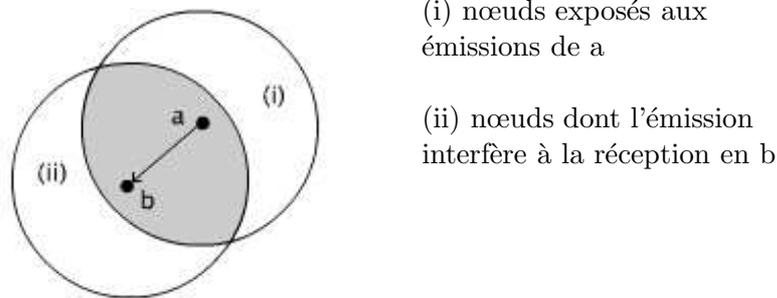


FIG. 2 – *Problème de la diffusion radio*

L'ensemble de ces nœuds (i) et (ii) sont sous la contrainte du flux ab, il sont en compétition d'accès au médium, en anglais contention<sup>1</sup>. Ceci nous mène à deux constatations :

- Un paquet de données peut avoir été perdu au niveau du receptr: l'acquie-temment n'est alors pas envoyé. Le nœud émetteur conclut à une collision et applique un *algorithme de résolution de contention* :

doubler la valeur de CW (en-deçà d'un certain maximum) et dès qu'une transmission est intégralement réussie, redonner à CW la valeur minimale

Il s'agit d'un schéma équivalent à celui d'Ethernet, et connu sous le nom de Binary Exponential Backoff (BEB). En cas de collision, il permet de diminuer leur nombre en augmentant la taille des fenêtres de tirage.

---

1. Nous utiliserons abusivement ce terme anglais

- Certains nœuds de la zone (ii) n'ont pas de moyen de se rendre compte de ces situations de compétition avec le flux ab: Le scénario est connu sous le nom de *problème des nœuds cachés* :

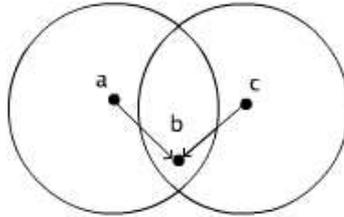


FIG. 3 – *Le problème des nœuds cachés*

Ci-dessus, le nœud c se trouve dans la zone (ii) du flux ab. Il ne devrait donc pas émettre. Cependant il est trop éloigné de l'émetteur a pour qu'ils s'autorégulent: le médium leur paraît constamment libre à tous deux malgré l'émission de l'autre. Le protocole les mène à augmenter leur fenêtre de contention (CW) à la valeur maximale, diminuant fortement leur débit et n'évitant pas la totalité des collisions en b.

Afin de régler le problème, la norme 802.11 spécifie l'ajout au cycle de transmission d'un envoi de paquets de réservation du médium, nommés Request-To-Send (RTS) et Clear-To-Send (CTS), et envoyés respectivement par le transmetteur et le récepteur.

Le RTS<sup>2</sup> est une demande d'allocation du médium et le CTS en est la réponse favorable *et* une injonction aux autres nœuds de rester inactifs (DEFER). Les paquets contiennent dans ce but un champ Duration<sup>3</sup> (durée) qui spécifie le temps d'occupation du médium – i.e.  $2 \times \text{SIFS} + \text{DATA} + \text{ACK}$ . Les chronogrammes dans le scénario des nœuds cachés sont donnés à la page suivante (Fig. 4)

---

2. Les RTS/CTS ne sont pas activés en-deçà d'une certaine taille du paquet de données.

3. Les nœuds gardent trace des différentes valeurs de Duration dans le NAV (Network Allocation Vector)

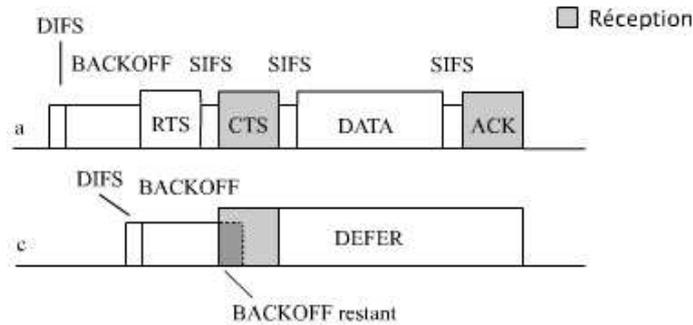


FIG. 4 – Le cycle de transmission avec les RTS/CTS

Le nœud *c*, touché par le CTS de *b*, ne peut débiter de transmission. Après lecture du CTS, il se place en attente DEFER, le mettant à égalité pour prendre le médium une fois que le nœud *a* a reçu l’acquiescement.

Ainsi la zone (ii) est protégée des interférences par les RTS/CTS<sup>4</sup>. Notons que l’ensemble de ces messages de signalisation font du médium radio une ressource très coûteuse en terme d’accès (temps d’accès double à celui du filaire).

## 2.3 Des interférences

On distingue deux zones de transmission, la portée de communication (TR, Transmission range) où tout signal est parfaitement traité et compris et la portée de détection (SR, Sensing Range) qui constitue l’espace où un nœud détecte de l’activité mais n’est pas en mesure d’en tirer de l’information.

Lorsqu’un nœud détecte de l’activité dans son SR, il considère le médium occupé. Il se place alors en mode de détection de porteuse (CS, Carrier Sense) avant de reprendre son cycle de transmission.

Dans la figure 5 le nœud *c* se trouve dans cette situation : il ne détecte que du bruit de la part de *a*. Par contre il est trop éloigné de *b* pour le détecter.

---

4. Il est clair que les collisions sont reportées aux RTS et là encore l’algorithme BEB est appliqué. Les paquets RTS étant de très petite taille leur réémission est moins coûteuse que celle des paquets de données: cf les études de Xu et al. [1]

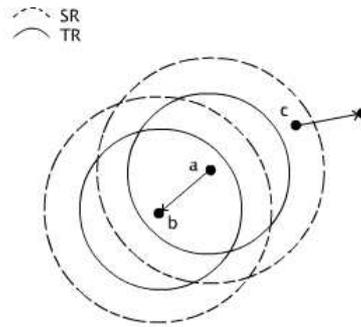


FIG. 5 – Une situation délicate

Or lorsque c se trouve dans la zone SR de l'émetteur, il n'a pas pu, à partir du RTS ou du DATA, conclure la valeur du champ Duration: il se positionne donc en CS et non en DIFFER, ce que l'on aperçoit sur les chronogrammes suivants.

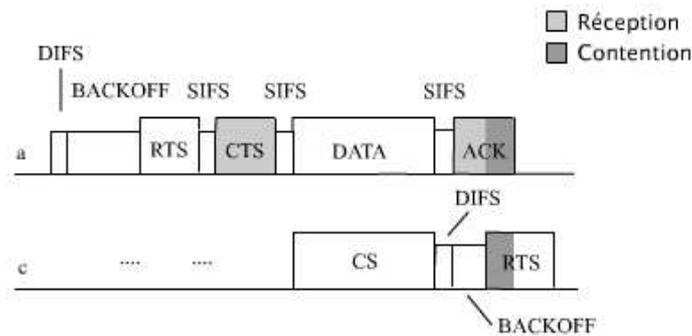


FIG. 6 – Chronogramme associé au scénario de la figure 5

On a montré ici le cas où c a réussi à débiter sa transmission – le médium lui apparaît en effet libre – alors que b émettait son acquittement. On a donc collision au nœud a<sup>5</sup>, repérée en gris foncé.

Il s'agit ici d'une faille qui oblige la norme à inclure un nouveau délai d'attente après chaque période CS, équivalent à SIFS+ACK+DIFS. Ce délai qui est nommé EIFS (Extended Inter Frame Space), permet la bonne réception des ACKs (ici en a) et de reporter le début du décompte du Backoff au même moment pour les deux nœuds concernés (a et c), afin que la prise du médium reste équitable.

5. Le signal de c perçu par a est bien sûr plus faible que celui perçu de b mais suffisant pour détériorer le paquet d'acquiescement. Certains définissent des cas – rares – où selon les distances le nœud est capable d'isoler le signal de b.

## 2.4 De la mobilité

Il est clair que les mécanismes de RTS/CTS et à fortiori d'EIFS peuvent se révéler totalement inefficaces si les nœuds bougent durant la transmission, ou que d'autres entrent dans le réseau, créant des interférences. L'exemple le plus simple est celui d'un nœud qui une fois qu'il a reçu un paquet de données s'éloigne trop de l'émetteur avant d'envoyer son acquittement.

Par la suite, on ne s'attardera pas sur la mobilité des nœuds mais sur des scénarii fixes où l'équité est discutée ou encore mise à mal. La description du protocole 802.11 étant terminée, il s'agit à présent de se questionner sur ce que l'on nomme équité.

## 3 Ce que devrait être l'équité

L'idée informelle d'équité est celle où un ensemble de nœuds en compétition pour le médium se partage la bande passante. Cette définition est une vision par nœud de l'équité. Cependant, on peut dans la situation suivante, à gauche, où tous les flux sont en compétition les uns avec les autres, vouloir que chaque flux détienne le même débit – throughput en anglais – plutôt que chaque nœud .

Signalons que 802.11 ne peut fournir d'autre vision que celle par nœud .

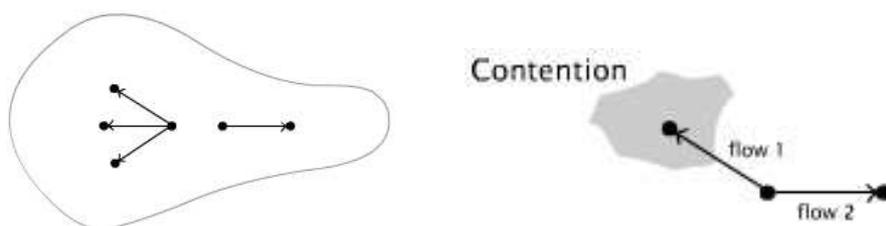


FIG. 7 – *Equité par flux*

Dans la partie droite de la figure 7, en supposant que le flux 1 souffre de contention (réception brouillée par exemple) il devient illogique comme dans 802.11, d'appliquer la même valeur de CW pour le flux 2, lui faisant perdre du temps et donc du débit.

Fort de cette dernière constatation, Bharghavan et al [2] proposent un protocole, MACAW, basé sur une application par flux de l'algorithme de Backoff. Celui-ci est d'ailleurs légèrement modifié: il double la fenêtre de contention CW à chaque collision à l'instar de BEB mais la décrémente à chaque paquet transmis avec succès, d'où son nom MILD (Multiplicative Increase Linear Decrease).

Nous retiendrons les deux idées suivantes:

- *le concept d'équité par nœud /par flux*. L'équité par flux paraît manquer de définition. Si l'on prend l'exemple de gauche (Fig 7) le flux de droite détient, à l'instar des autres, un quart du médium. Si l'on suppose l'existence d'un cinquième flux qui ne soit en compétition qu'avec le flux de droite, pourquoi

se bornerait-il à ne prendre lui aussi qu'un quart du médium quand il peut en prendre trois fois plus? La vraie question est de savoir comment disséminer de telles informations. Le choix entre ces deux visions de l'équité devrait être dépendante tant de la topologie que de l'application que l'on souhaite avoir de son réseau.

- *la fenêtre de contention doit être le reflet de la compétition subie par un flux et non un nœud* .

Dans ce sens, Ozugur et al [3] proposent le protocole CB-fair qui utilise la persistance. La persistance est la probabilité qu'une fois le Backoff expiré, la transmission s'effectue. Notons que 802.11 la spécifie à 1. Pour les chercheurs elle peut réguler les situations de compétition. Ainsi il propose d'appliquer un taux proportionnel au ratio du degré (au sens des flux) du récepteur sur le degré maximum des voisins du transmetteur, données qui sont diffusées à travers le réseau. Là encore l'algorithme de Backoff est modifié lorsqu'une transmission est sans erreur : on divise par deux la fenêtre de contention CW.

On retiendra surtout que l'on peut réguler les envois autrement qu'en modifiant simplement le paramètre CW. En effet la persistance est certainement le meilleur moyen de modéliser une allocation du médium faite à un nœud ou à un flux et ainsi éviter les compétitions d'accès.

Nos deux papiers précédents cherchent à modifier l'algorithme d'ajustement de la fenêtre de contention (CW), obtenant d'ailleurs de meilleurs résultats. En effet, l'algorithme BEB se résume en *favoriser la dernière station ayant réussi une transmission*. Il est évident que ceci va à l'encontre d'une station qui n'arriverait pas à transmettre. Une étude combinatoire sur les chargements du paramètre CW permettrait de dire s'il doit être réellement stabilisé ou fluctuant, mais aucune recherche n'a été menée sur ce sujet.

Il est à noter que les protocoles MACAW et CB-FAIR font figure de référence pour évaluer les performances de nouveaux protocoles. Ils améliorent sensiblement l'équité dans plusieurs scénarios par rapport au protocole de 802.11. Cependant nous allons nous intéresser à présent à ce qu'ils ne résolvent pas et à exposer un point de vue plus large du problème.

## 4 Ce qu'est l'inégalité

### 4.1 Un scénario simple

Voici ci-dessous un scénario irrésolu par les protocoles vus précédemment. Il s'agit de deux flux, avec le récepteur de l'un en portée de communication avec l'émetteur de l'autre. Certains parleront de nœud caché (0&2); d'autres de nœud exposé (1).

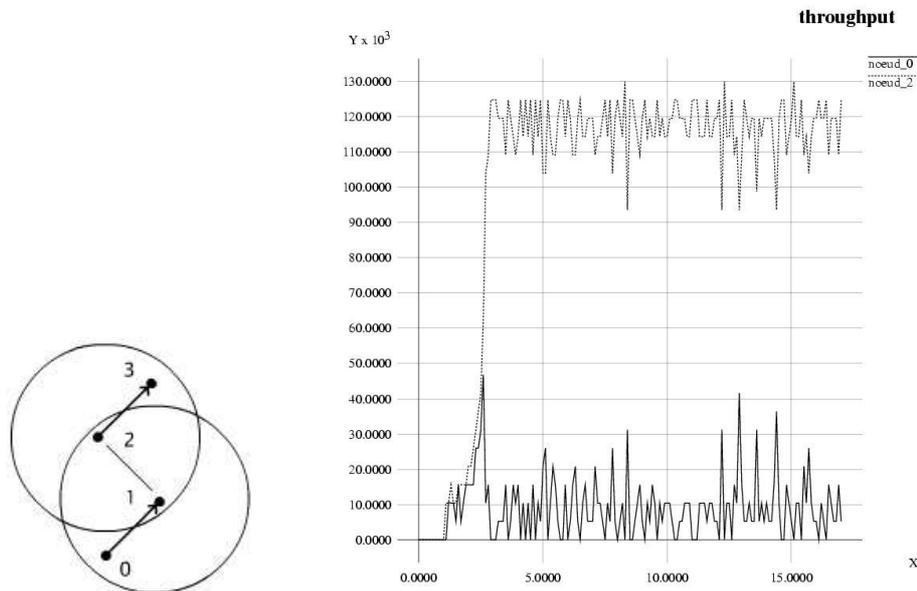


FIG. 8 – *Scénario nœuds cachés & exposés : résultats*

A droite, le graphique<sup>6</sup> représente le débit des nœuds 2 (courbe haute) et 0 au cours du temps (première phase : montée du débit, puis à 3s. débit maximum) sous 802.11.

Le nœud 2 lèse complètement le nœud 0. L'explication tient dans le fait que pour le nœud 2 toutes ses transmissions réussissent. Il reçoit immédiatement le CTS et l'ACK du nœud 3. Le nœud 0 est dans la situation inverse. Le nœud 1 étant exposé au nœud 2, celui-ci est constamment le siège de collisions – puisque le nœud 2 étant caché au nœud 0, ce dernier envoie des RTS. Le nœud 1 ne reçoit donc aucun des RTS de 0. Il ne renvoie donc aucun CTS plaçant le nœud 0 en situation de famine. Conséquence désastreuse de l'algorithme BEB – qui s'applique ici du au non retour de CTS –, la fenêtre de contention CW du nœud 0 reste bloquée à la borne maximum écartant toute chance d'accès au médium.

Notons surtout que le nœud 2 n'a aucun retour d'information sur le problème. Il occupe le médium en toute bonne foi. Cette simulation nous a conduit à revoir le problème d'un point de vue plus global. C'est ce que traite le paragraphe suivant.

## 4.2 Approche théorique de l'équité

Il s'agit de proposer un modèle qui capture tous les cas de compétition. On définit un lien radio entre deux nœuds lorsque ceux-ci sont dans leur zone TR. Ces nœuds sont dits voisins (TR-voisinage). On parle aussi de SR-voisinage.

6. Dans toutes les simulations qui ont été faites les nœuds travaillent au débit maximal. Les simulations ont été réalisées sous le simulateur NS-2 (cf Appendice).

Soit un flux  $\mathcal{F}$ . Si l'on se souvient des zones (i)(ii), définies dans la Fig. 2, un certain nombre de contraintes sont posées autour du flux  $ab$ . De plus, les zones s'inversent lors de l'émission des paquets CTSs/ACKs. Ainsi l'ensemble des flux qui mettent en jeu des nœuds se trouvant dans l'une de ces deux zones rentrera indubitablement en compétition avec le flux  $ab$  pour l'accès au médium. La Fig. 9 résume de façon schématique l'ensemble des flux en compétition avec  $\mathcal{F}$ .

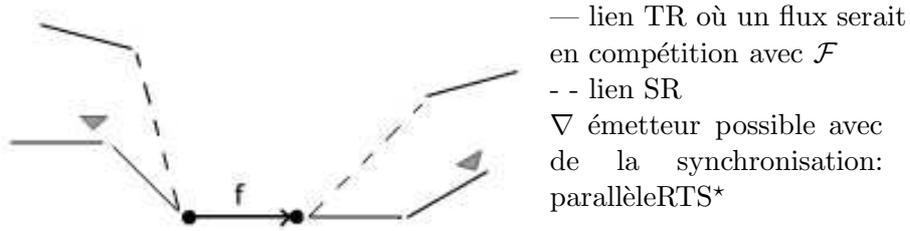


FIG. 9 – Dessin synoptique des flux en contention avec  $\mathcal{F}$

Le protocole 802.11 est bien loin de toutes ces considérations. Par contre, pas de celles de Bharghavan [4] qui propose de construire un *flow contention graph*  $\mathcal{G}$  où les nœuds sont les flux du réseau et où les arêtes symbolisent l'existence de compétition entre eux.

La Fig. 10 donne un exemple de réseau et le graphe  $\mathcal{G}$  correspondant. La première information qu'il fournit est que dans une même clique, un seul flux peut être utilisé.

Cependant une dernière étape de modélisation est nécessaire afin d'élire correctement les flux qui peuvent être actifs simultanément. Il s'agit de construire le *ressource graphe*, graphe biparti avec d'un côté des nœuds représentant les cliques maximales de  $\mathcal{G}$  appelées serveurs et de l'autre côté les flux. On a alors une arête ssi le flux appartient à un serveur dans  $\mathcal{G}$ .

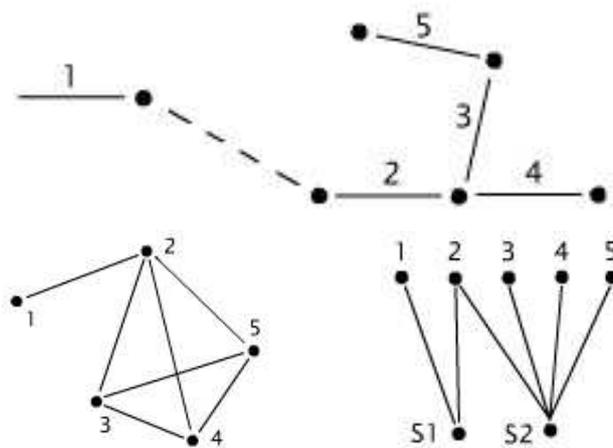


FIG. 10 – Dans l'ordre: exemple, flow contention graph  $\mathcal{G}$  et ressource graph

Ainsi un protocole simple évitant toute situation de compétition pourrait être d'utiliser un système de jetons. Les serveurs disposent chacun d'un jeton et un flux peut être actif ssi il a reçu le jeton de chaque serveur auquel il est incident.

Ceci étant il reste à définir :

- une fonction modélisant l'allocation faite à chaque flux – i.e. la distribution des jetons. A partir de la fonction, on constitue un vecteur d'allocation  $\mathcal{V}$  qui contient le taux de ressources allouées à chaque flux.
- à quel point on peut utiliser les outils précédemment définis. En effet, trouver les cliques d'un graphe est un problème NP-complet. De plus cela suppose que chaque nœud ait une connaissance de la topologie du réseau<sup>7</sup> dans l'espace que l'on a défini à la Fig. 9. Une telle connaissance est synonyme de paquets de signalisation nombreux, saturant le réseau.

**Algorithme résultant** Les mêmes chercheurs proposent d'appliquer  $\mathcal{V}$  à l'aide de la persistance et que la fenêtre de contention CW soit unique à l'ensemble des flux d'une clique puisqu'ils sont tous en compétition pour utiliser un même médium. Ils donnent un cadre général pour établir le vecteur  $\mathcal{V}$  qui bien sûr dépend du graphe  $\mathcal{G}$ . De nombreux chercheurs [5] se sont penchés sur le critère d'équité max-min afin d'allouer  $\mathcal{V}$ . Les résultats sont difficilement évaluables. De plus des études menées per l'EPFL [6] ont montré que dans des scénarios simples il n'était pas souhaitable d'un point de vue de l'équité, d'utiliser ce critère d'allocation.

L'algorithme final que propose Bharghavan ne tient pas compte du graphe  $\mathcal{G}$  pour les raisons invoquées au paragraphe précédent. Le vecteur d'allocation évolue selon les pertes que chaque flux subit. Les résultats sont sensiblement meilleurs que ceux de CB-FAIR et MACAW mais les simulations exposées mettent toujours en jeu des réseaux denses où certains problèmes peuvent être gommés, comme nous allons le voir au chapitre suivant.

## 5 Une inégalité protocolaire

Fort de ces résultats sur les compétitions d'accès, nous nous sommes demandé si l'inégalité ne pouvait pas apparaître sous d'autres aspects dans 802.11.

Le système de l'EIFS – cf section 2.3 – nous a alors interpellé. En effet, l'EIFS correspond à une attente obligatoire et sa condition de déclenchement – après tout passage en détection de porteuse CS – ne paraissait pas rendre compte de tous les cas possibles.

Ci-dessous deux scénarios, où le flux 01 est en mouvement – 0 vers 0'. Les arcs en pointillés dénotent la zone SR du nœud 3; les liens en pointillés indiquent que deux nœuds sont dans leurs zone SR; les liens pleins, dans leur zone TR.

---

7. On ne voit pas vraiment comment un nœud pourrait construire le graphe  $\mathcal{G}$  lorsque des liens SR sépare des flux en compétition. Par exemple les flux 1 & 2 de la Fig. 10

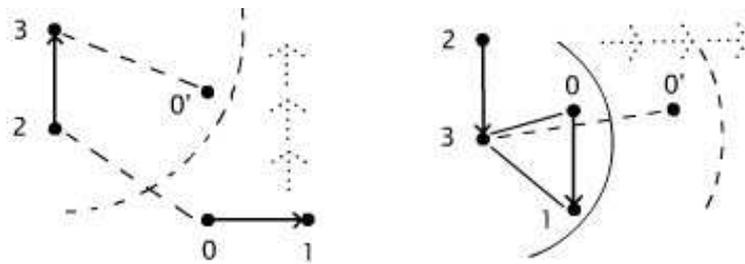


FIG. 11 – Les deux problèmes de l'EIFS

**Scénario de gauche** *réalisé sans RTS/CTS, 0 en 0'*. Supposons que 0 soit en CS dû au DATA transmis par 2. Il s'en suit un premier EIFS qui va être stoppé par la transmission de l'acquitement de 3. En effet comme 0 se trouve également dans la zone SR de 3 quand 0 est en 0', il se retrouve en CS. Automatiquement un second EIFS est ajouté, qui est inutile puisque la réception de l'ACK de 3 en 2 a déjà été protégée (comme indiqué dans le chronogramme figure 12). L'EIFS devient alors une pénalité pour le nœud 0 qu'il subit à chaque fois qu'il souhaite débiter un cycle de transmission. La prise du médium est inéquitable.

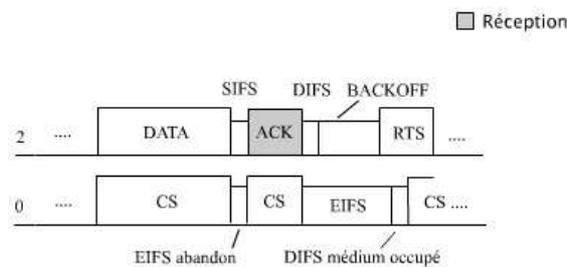


FIG. 12 – Large EIFS: chronogramme – sans RTS/CTS

Sur la partie gauche de la figure 14 (graphique des débits de 0 et 2), lors des 5 premières secondes, le nœud 0 est encore au-delà de la zone SR de 3, l'équité est visible. Puis le nœud 0 rentre dans la zone SR de 3, le nœud 2 prend alors le pas sur le nœud 0.

**Scénario de droite** *réalisé avec RTS/CTS, 0 en 0'*. Il s'agit du même problème reporté sur le CTS. Lorsque le nœud 0 se trouve dans la zone SR de 3, il va se placer en CS lorsque 3 va émettre un CTS. Cette fois, l'EIFS étant plus court que le délai SIFS+DATA. Le nœud 0 pourrait être à même de commencer une transmission, puisqu'il n'a pas connaissance du nœud 2 et n'a pas pu tirer d'information du CTS. Si c'est ce qui se produit, deux collisions peuvent survenir (Fig 13). Une première avec le DATA de 2 au nœud 3. Si celle-ci n'a pas eu lieu (le RTS est émis pendant le SIFS qui suit le DATA) une seconde a lieu avec l'ACK de 3 au nœud 1.

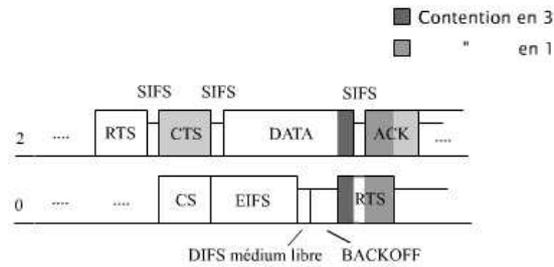


FIG. 13 – *Small EIFS: chronogramme – avec RTS/CTS*

L'interprétation du graphique (partie droite de la figure 14 est plus difficile. La première partie correspond au moment où 0 est encore dans la zone TR de 3 (on se retrouve dans la configuration de la figure 8), occupant tout le médium. Puis 0 passe dans la zone SR de 3, cas que l'on vient de décrire. On remarque que son débit devient nul<sup>8</sup>. N'accédant pas au médium sa fenêtre de contention CW est au maximum. Enfin le nœud 2 émet mais à 60 % en dessous du débit maximal, confirmant l'analyse: le nœud 0 envoie toujours des RTS qui brouillent la réception au nœud 3.

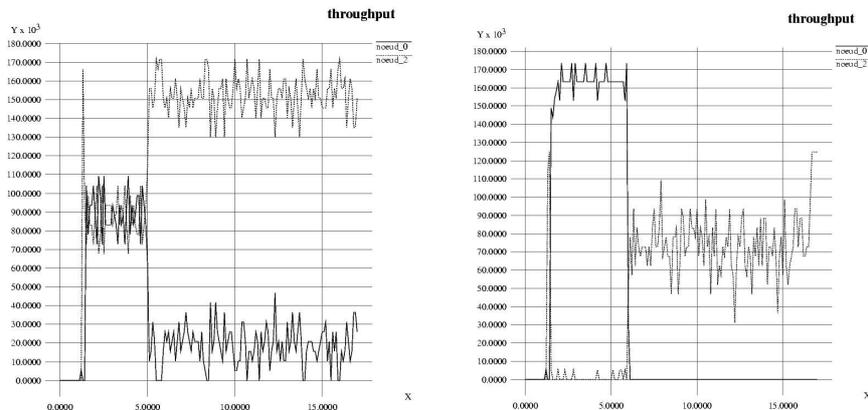


FIG. 14 – *EIFS: résultats des simulations*

Le problème a été reconnu également par Li et al. [7] dans d'autres configurations. Ils nomment le premier problème large-EIFS puisque l'EIFS est trop long par rapport à ce qu'il devrait être et le second small-EIFS puisque trop court pour rendre réellement compte de l'occupation du médium.

Ils proposent une solution basée sur l'idée que dans la zone SR, selon le temps passé en CS, on peut conclure si on se trouve dans l'une ou l'autre des deux situations. Une technique difficile à mettre en oeuvre et peu fiable.

La solution que nous proposons dans le dernier chapitre semble résoudre le problème.

<sup>8</sup>. On ne parlera plus d'inégalité mais d'iniquité!

## 6 Une inégalité intrinsèque

Au sein de notre laboratoire Dominique Dhoutaut [8] a mis à jour un scénario capable de déjouer bon nombre de protocoles. La simulation sera donc encore faite sous 802.11. Il s'agit du problème des *trois paires* nommé ainsi puisqu'il met en jeux trois flux parallèles.

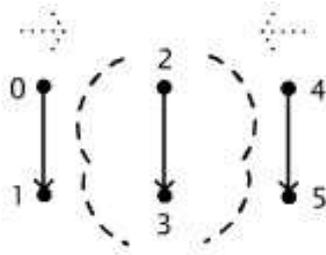


FIG. 15 – *Le scénario des trois paires*

Lorsque le flux central se trouve dans les zones SR des flux extérieurs, il souffre des problème de l'EIFS que nous venons de décrire. Cependant, même en suposant ce problème résolu, un second facteur intervient: la synchronisation. En effet, les deux paires extérieures ne sont pas obligées d'émettre en même temps, c'est-à-dire d'avoir leurs cycles de transmission en phase.

En effet, les nœuds 0 et 4 peuvent émettre de telle façon que leurs paquets DATA se chevauchent. Le chronogramme (Fig. 16) retrace les activités des nœuds lorsque 2 se trouve dans les zones SR des deux nœuds extérieurs. Le nœud 2 ne peut que se placer en détection de porteuse (CS), *ad vitam et aeternam*, sans même avoir la possibilité d'émettre un RTS.

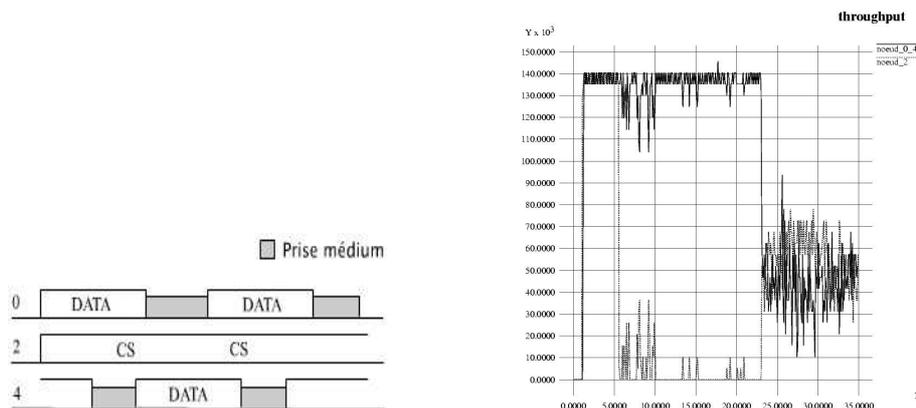


FIG. 16 – *Les trois paires: chronogramme & résultats*

Le graphique des débits montre que jusqu'à 5s, les nœuds 0, 2 et 4 ne se détectent pas. De 5s. à 23s. les nœuds 0 et 4 sont dans la zone SR du nœud 2. Cependant si l'inégalité apparaît instantanément elle n'est que plus franche à partir de la seconde

10. L'explication tient dans le fait que les flux extérieurs ne sont pas parfaitement désynchronisés, permettant au nœud 2 d'envoyer quelques RTS qui n'auront pas de réponse dû au problème de l'EIFS. La fenêtre de contention CW du nœud 2 grimpe à la valeur maximale. Son débit restera presque nul jusqu'à ce que les nœuds 0 et 4 arrivent dans la zone TR de 2, moment où l'équité est respectée.

Le problème des trois paires est le plus ennuyeux de tous et à deux titres:

- il ne paraît pas être un motif réseau rare pour le ad-hoc (exemple d'avions en formation...)
- les trois paires ne peuvent pas communiquer entre elles lors de la phase problématique<sup>9</sup>.

La solution proposée par le chercheur est la suivante: les flux infligent par eux-mêmes un malus à leur fenêtre de contention CW selon le nombre de paquets qu'ils ont envoyés avec succès. Une courbe de malus est appliquée (premier plateau faible pour les premiers paquets puis une ascension exponentielle avant d'atteindre un plateau maximal). Cependant c'est un mécanisme qui peut s'avérer gênant quant au débit dans des scénarios où de tels problèmes n'existent pas.

Ainsi, il s'agit bien d'un problème intrinsèque au médium radio, puisque des nœuds peuvent être en compétition avec d'autres nœuds sans qu'aucune communication claire soit possible entre eux sur ce même médium pour résoudre le problème.

## 7 Une solution

Il n'y a pas de recherche de solutions sans établir préalablement de bilan. Il est à noter que notre compréhension de la couche MAC des réseaux ad-hoc n'a pas suivi le même déroulement dans ce rapport que pendant le stage.

Celui-ci avait au programme la mise en place de protocoles récents. Nous avons choisi d'implémenter sous NetworkSimulation NS un algorithme de résolution de fenêtre de contention CW, proposé par M. Bensaou [9], afin de l'évaluer.

L'une des conclusions qu'on peut tirer des chapitres précédents est que le graphe de compétition  $\mathcal{G}$  ne saisit pas la totalité des problèmes de l'équité, que ce soit dans le cas de notre premier scénario (Fig. 8) ou dans le cas des problèmes de l'EIFS. Tous ces flux appartiendraient à une même clique et à contrario de ce que proposait Bharghavan, ces flux ne sont pas tous identiquement dans la même situation et doivent avoir un paramètre CW différent. Il est donc capital de mettre au point un algorithme performant pour décider de sa valeur.

L'idée première de l'algorithme de Bensaou est que chaque nœud effectue un comptage des paquets (en terme de temps d'occupation du médium) qu'il envoie, correspondant à  $W_i$  et qu'il reçoit correspondant à  $W_o$ ).

Une métrique est alors définie, l'index d'équité (FI, Fairness Index) égal à  $\frac{W_i}{W_o} \times \frac{\phi_o}{\phi_i}$ ,

---

9. Il est alors intéressant d'utiliser un protocole de routage qui recherche des routes entre les mobiles avant d'effectuer l'envoi des paquets. Dans ce scénario, le nœud central n'aura pas le temps de les construire, empêchant tout envoi de paquet

où  $\phi_i$  est l'allocation faite au nœud  $i$  et  $\phi_o$  la somme de celles faites à ses voisins émetteurs (ensemble noté  $\Gamma$ ).

Nous allons nous intéresser au cas où l'allocation de tout un chacun est de 1. On a alors  $FI = \frac{W_i}{W_o} \times |\Gamma|$ .

Bensaou propose alors que CW soit réajusté de la manière suivante: chaque nœud double/divise de moitié sa fenêtre de contention selon si son index d'équité est supérieur/inférieur à 1.

**Résultats** Dans leur rapport de recherche sont exposés les résultats obtenus avec le scénario décrit à la Fig 8, que nous nommerons A. Nous avons implémenté leur algorithme sous NS à la place de celui de 802.11 et, en effet, le débit des deux mobiles O & 2 est équitable bien qu'ils n'occupent à eux deux que 40% du médium. Cependant, en regardant l'évolution du paramètre CW au cours du temps, pour le nœud 2, nous nous sommes rendus compte que l'algorithme ne l'avait conduit qu'à stagner à la valeur maximale et de façon immédiate. Nous pouvons tirer quelques conclusions essentielles sur leur algorithme:

- sous 802.11, le paramètre CW se situe dans l'intervalle [32..1024]. En utilisant encore un schéma exponentiel, il suffit de cinq réévaluations pour passer d'une borne à l'autre. Non seulement les changements sont brutaux mais de plus, ils ne sont pas limités dans le temps afin de voir si le changement de paramètre a été profitable ou non.
- dans le scénario A, les nœuds émetteurs 0 et 2 ne se détectent pas. Ils n'ont aucun moyen de faire un comptage des paquets que l'autre envoie. C'est certainement pour cette raison que le comptage des paquets effectué dans l'algorithme nous est apparu anarchique (paquets comptés plusieurs fois).
- enfin les graphiques produits quant à l'évolution de l'index d'équité FI montrent que celui-ci se stabilise à des valeurs incohérentes puisque fluctuantes entre 2 et 4. Chaque nœud devrait pouvoir la stabiliser à 1.

## 7.1 Un algorithme de résolution de fenêtre de contention

Nous n'avons cependant pas écarté l'idée d'espionner le médium. De plus l'avantage d'une telle solution est qu'elle s'intègre au protocole 802.11. Notre algorithme se trouve annexé à ce rapport. Nous allons à présent décrire ses deux phases, le comptage des paquets et l'ajustement de la fenêtre de contention.

**Espionnage** le but est que chaque nœud détienne une estimation précise de  $W_o$  en bits/s. Afin d'effectuer un comptage réaliste il faut prendre en compte tous les nœuds qui sont en compétition. Nous allons nous baser sur la figure suivante (Fig. 17), qui remémore les zones (i) et (ii), pour dresser l'ensemble des données à récolter et comment les obtenir.

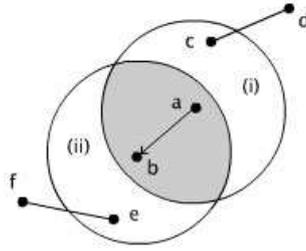


FIG. 17 – *Compétition entre les nœuds : aide au calcul de  $W_o$*

- tout nœud dans (i) touche directement a, donc pas de difficultés à l'évaluer.
- tout nœud dans (i) est exposé. Si un nœud hors de (i), par exemple d, émet vers lui, il est en compétition avec a. Cependant a ne le détecte pas: les émissions de ce nœud seront évalués à travers les CTSs et ACKs du nœud exposé c qui lui touche a (aux CTSs/ACKs qu'il reçoit de c, il conclura au nombre de RTS/DATA que d émet).
- tout nœud dans (ii)\(i), par exemple e, est en compétition avec a, mais là encore a ne le détecte pas. On pourrait appliquer le principe précédent s'il émettait vers un nœud dans (i). Si ce n'est pas le cas (il émet vers f par exemple) l'unique solution est d'utiliser un nœud pour relayer l'information. Nous avons choisi d'ajouter ces informations de comptage lors de l'envoi du CTS (ou de l'ACK si les RTS/CTS sont désactivés) par le nœud vers lequel on émet, ici b.
- tout nœud émettant vers un nœud situé en (ii), f par exemple, est en compétition avec a. Là encore c'est le nœud b qui relayera l'information, qu'il aura trouvée en utilisant la même méthode qu'au deuxième point – par les émissions de CTSs et acquittements de e.

Ainsi le nœud b va communiquer dans son CTS son estimation de  $W_o$ . Celle-ci inclura autant le nœud a lui-même que d'autre nœud dont a a déjà fait le compte. Ceci nous contraint à ce que les nœuds effectuent un comptage séparé pour chaque nœud l'environnant. Nos paquets CTS détiendront alors la liste L des nœuds estimés<sup>10</sup> et la *somme* de ces estimations S. Afin d'évaluer son propre  $W_o$ , le nœud a n'a plus qu'à soustraire à S sa propre estimation des nœuds qu'il connaît (en corrélation avec L), dont lui-même. La cardinalité de L lui permet enfin de connaître le nombre de nœuds avec lequel il est en compétition  $|\Gamma|$ .

**Réajustement de CW** A partir des données de comptage, le calcul de FI intervient ainsi que celui de trois données importantes, qui sont l'évolution de FI,  $W_i$  et  $W_o$  depuis le dernier changement de la valeur de CW.

---

10. Celle-ci peut-être conséquente. Il serait souhaitable qu'elle ne soit envoyée qu'une fois, et que par la suite seules les entrées/sorties de cette liste soient intégrées au paquet CTS.

Celui-ci ne peut pas avoir lieu que sous trois conditions (On se place dans le cas où le nœud lèse ses voisins, ie  $FI > 1$ , l'autre cas est symétrique).

- FI doit dépasser 1 augmenté d'une tolérance que l'on fixe.
- la période d'adaptation à la valeur de CW actuelle doit être terminée, c'est à dire, cela fait plus de n (paramètre de l'algorithme) paquets reçus/envoyés, et de façon consécutive que FI est supérieur à 1.
- l'évolution de FI doit être supérieure à 0.9, ceci afin de rendre compte si la situation s'est dégradée (avec 10% de tolérance) ou améliorée.

Ces conditions remplies, la valeur de CW est modifiée en la multipliant à la fois par l'évolution de FI et de celle de Wi. On assure ainsi que l'augmentation de la fenêtre de contention est équilibrée et prend en compte l'évolution propre du débit du nœud . De plus il n'est pas appliqué de facteur de multiplication supérieur à 2.

## 7.2 Résultats

Nous avons sélectionné ici deux de nos simulations. La première n'est autre que celle du scénario A. La seconde simule un scénario où le problème de l'EIFS se produit. Pour chacune les graphiques de l'évolution du débit, de CW et de FI au cours du temps sont donnés.

**Scénario 1** le résultat quant au débit est très positif, puisque non seulement l'équité est respectée à contrario de ce que nous avons vu sous 802.11 (page 8) mais en plus les variations de débit subies par les nœuds sont faibles. L'index d'équité fait preuve de bonne stabilité.

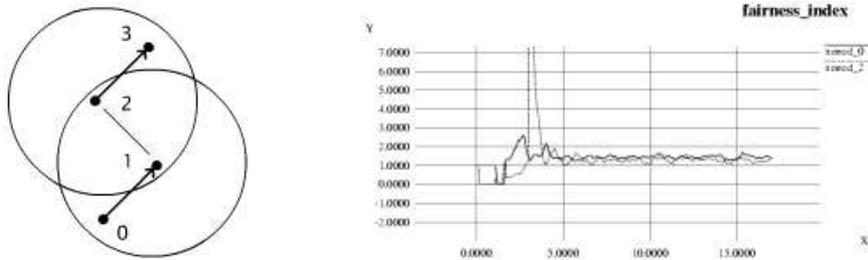


FIG. 18 – Scénario 1 – évolution de FI au cours du temps

Quant à la variation de la fenêtre de contention, nous avons bien celle du nœud 2 plus haute que celle du nœud 0. Cependant elles sont anormalement hautes toutes les deux, ce qui se traduit par une perte du débit global (65% du médium occupé). Le problème vient du fait qu'au début de la simulation peu de données sont récoltées. Un paquet envoyé par l'un peut peser lourd dans l'évaluation de l'autre. Une fois cette période passée on acquiert une bonne stabilité du paramètre CW, mais bloqué à de trop hautes valeurs.

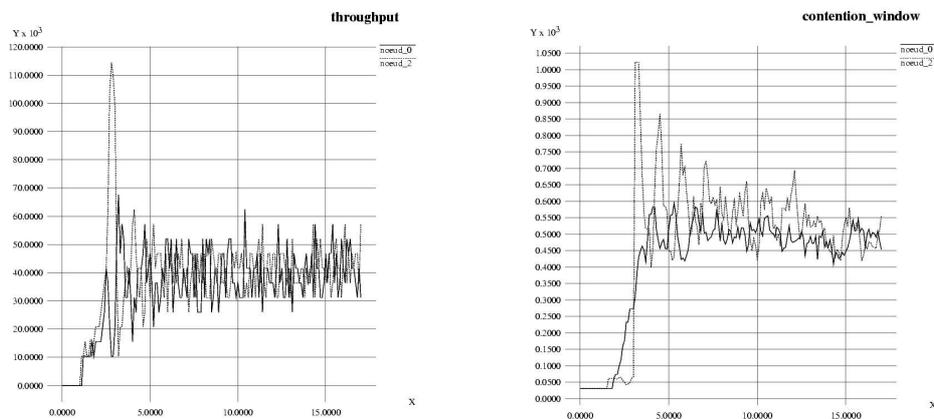


FIG. 19 – Scénario 1 – débit à gauche et évolution de CW à droite

**Scénario 2** il s’agit des deux mêmes flux placés de telle sorte que tous soient à portée l’un de l’autre sauf les nœuds 0 et 3 qui se trouvent dans leur zone SR. Le nœud 0 souffre alors du problème large-EIFS lorsque le nœud 3 émet ses acquittements. Notons que sans ce problème ce scénario est traité équitablement par 802.11. Le graphique des débits montre à la fois ceux obtenus sous 802.11, où l’inégalité est manifeste, et ceux obtenus à l’aide de notre algorithme. Le débit total est encore amélioré par rapport à notre exemple précédent.

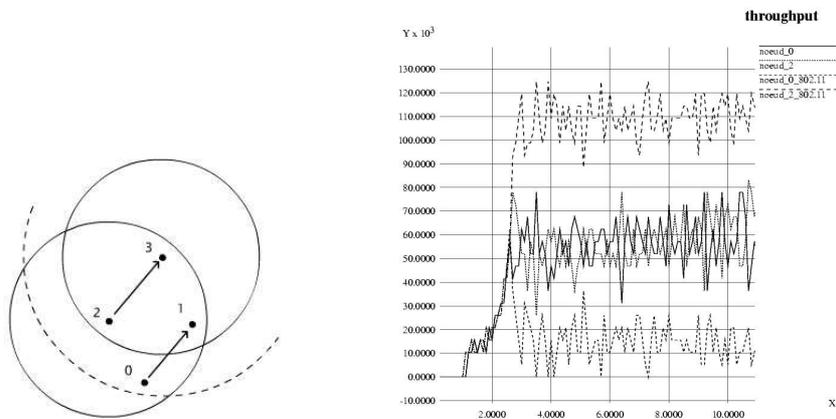


FIG. 20 – Scénario 2 – débits en regard de ceux expérimentés sous 802.11

La courbe la plus surprenante est celle de la fenêtre de contention. Pour les mêmes raisons évoquées ci-dessus, les valeurs de CW commencent par augmenter fortement. Cependant elles gardent entre elles un écart significatif, que l’on peut donner comme la compensation de l’EIFS subit par le nœud 0 ! La courbe de l’index d’équité ne nous donne pas autant de satisfaction. Le fait que le nœud 2 évalue qu’il détient un débit double à celui de 0 provient du fait qu’il

prend en compte les émissions de 1 dans son calcul de  $W_o$ , ce que ne peut pas faire le nœud 0 avec le nœud 3.

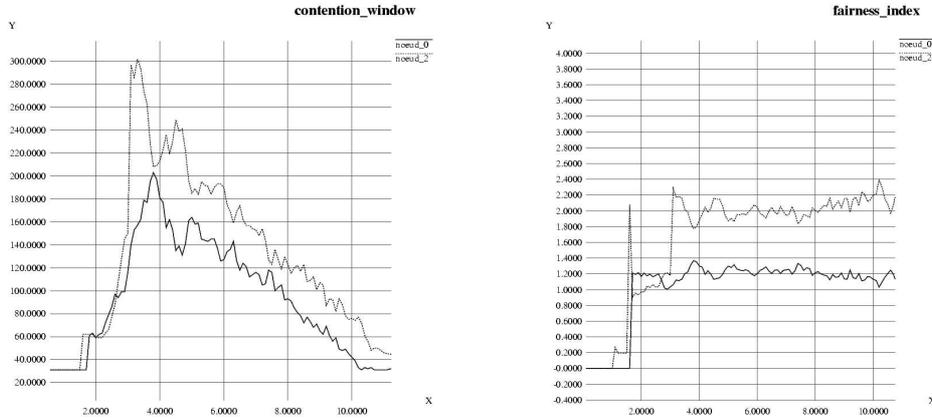


FIG. 21 – Scénario 2 – évolution de CW à gauche et de FI à droite

Ainsi l’algorithme que nous proposons c’est montré capable de résoudre le problème du nœud exposé ainsi que celui de l’EIFS.

Il reste que le débit global est parfois malmené, cependant moins qu’avec l’algorithme de Bensaou, qui d’ailleurs s’est révélé inefficace dans le cas du deuxième scénario. Les protocoles MACAW et CB-FAIR ne résolvent pas non plus ces difficultés.

Il ne faut pas se voiler les yeux. il s’agit d’une approche prometteuse mais qui nécessite que soient réalisées des études combinatoires, ou encore de prouver formellement ses bienfaits. Il faut également trouver un moyen de contenir l’expansion de CW au prémices de l’algorithme.

## 8 Conclusion

Ainsi le problème de l’équité dans les réseaux ad-hoc au niveau de la couche MAC part du protocole 802.11, le standard actuel, qui ne donne que peu de garantie en la matière.

Si les chercheurs ont d’abord éclairé ce que devait être l’équité et les emplois de la fenêtre de contention et de la persistance, ils n’ont pas à travers leurs études théorique de la compétition au sein d’un réseau, capturé les problèmes réels que rencontre le partage du médium radio.

L’algorithme que nous avons proposé donne des résultats très satisfaisants tant pour les problèmes de nœuds exposés que ceux dus à l’EIFS. Par là même, il affirme l’utilité d’un algorithme réfléchi de résolution de fenêtre de contention.

Inéluctablement les chercheurs devront s’atteler à mieux modéliser l’ensemble des phénomènes engendrés par le médium radio. La complexité de celui-ci les entrainera certainement à se tourner vers d’autres solutions telles que les antennes intelligentes – qui peuvent restreindre leur espace d’émission – ou encore remettre fondamentalement en cause les protocoles d’accès au médium actuels.

## Références

- [1] Kaixin Xu, Mario Gerla, Sang Bae: *How effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks?*, University of California
- [2] V. Bharghavan, A. Demers, S. Shenker, L. Zhang: *MACAW: A Media Access Protocol for Wireless LANs*, ACM SIGCOMM 1994
- [3] T. Ozugur, M. Naghshineh, P. Kermani, C. Olsen, B. Rezvani, J. Copeland: *Balanced Media Access Methods for Wireless Networks*, ACM MOBICOM 1998
- [4] Vaduvur Bharghavan, Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao: *Achieving MAC Layer Fairness in Wireless packet Networks*, Coordinated Science Laboratory, University of Illinois
- [5] Brahim Bensaou, Xiao Long Huang: *On Max-min Fairness and Scheduling in Wireless Ad-Hoc networks*, Hong-Kong University of Science and Technology
- [6] Bozidar Radunovic, Jean-Yves Le Boudec: *Why Max-min Fairness Is Not Suitable For Multi-Hop Wireless Networks*, Ecole Polytechnique Fédérale de Lausanne
- [7] Zhifei Li, Sukumar Nandi, Anil K. Gupta: *Improving Fairness in IEEE 802.11 based MANETs using Enhanced Carrier Sensing*, University of Singapore
- [8] Dominique Dhoutaut, Isabelle Guerin-Lassous: *Impact of heavy traffic beyond communication range in multi-hops ad-hoc networks*, Ecole Normale Supérieure de Lyon
- [9] Brahim Bensaou, Yu Wang: *Fair Medium Access in 802.11 based Wireless Ad-Hoc Networks*, Centre for Wireless Communications, National University of Singapore – Mobihoc 2000
- [10] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci: *Wireless Sensor Networks: A Survey*, Computer Networks, 38(4):393-422, March 2002.
- [11] Ad Hoc Mobile Wireless Networks, *book* Protocols and Systems de C-K Toh, éd Prentice Hall.
- [12] Aravind Velayutham, Haoli Wang: *Solution to the Exposed Node problem of 802.11 in Wireless Ad-Hoc Networks*, Departement of Computer Science, Iowa State University Ames

## A Curiosités

Cette première partie de l'appendice regroupe quelques problématiques et termes qui touchent aux réseaux ad-hoc. Ils ont été repérés dans le rapport à l'aide d'une étoile \*. Ces notions ont été abordées lors de conférences ou de lectures solitaires. Ce sont donc des pistes à suivre pour s'immerger dans le monde des réseaux ad-hoc.

**Réseaux Hybrides** (p.3) – l'exemple qui nous intéresse, est celui d'un réseau ad-hoc relié à des stations de base, elles-mêmes reliées à l'internet filaire. Il s'agit d'élire au sein du réseau ad-hoc des clusters de nœuds avec en eux des leaders qui vont relayer l'ensemble des données aux stations de base. Il existe encore de nombreux problèmes comme par exemple, l'adressage IP au sein du réseau ad-hoc.

**Sensor-Network** (p.3) – ce sont des réseaux de capteurs à l'étude surtout dans le domaine militaire. L'idée est de déployer sur un espace étendu, un réseau dense de capteurs (chimiques, etc) et à l'aide par exemple d'avions survolant le réseau, de récolter les informations. Les deux problématiques essentielles sont la récolte des données au sein du réseau (les données sont d'abord routées vers certains nœuds) et l'optimisation de l'énergie, donc du temps de vie de chaque capteur. Les nœuds responsables de l'envoi des données au moment de la collecte seront les premiers à ne plus avoir d'énergie. Il est par exemple proposé qu'au déploiement du réseau, les capteurs aient des autonomies différentes (cf le rapport de recherche [10]).

**Protocoles de Routage** on distingue deux classes : les réactifs et les proactifs. Les premiers cherchent une route en inondant le réseau (i.e. tout nœud qui reçoit le message le retransmet) d'un message de recherche du destinataire. Un message retour est envoyé explicitant la route créée. La seconde approche utilise des tables de routage mais nécessite leur actualisation. Souvent, on définit dans le voisinage d'un nœud, l'ensemble des nœuds qui lui permettent d'accéder à son 2-voisinage afin d'optimiser les recherches et envois multiples de messages. En résumé, de beaux problèmes: cf [11] pour approfondir.

**Parallèle RTS** (p.12) – On voit que les deux nœuds marqués par  $\nabla$  (fig 9) pourraient en effet transmettre en parallèle au flux  $\mathcal{F}$  si leurs cycles de transmission étaient en phase – par exemple pour le flux noté à gauche, les émissions vont s'interférer aux émetteurs des flux, ce qui n'est pas gênant, mais pas les réceptions des CTSs/ACKs puisqu'elles interviendront au même moment. Pour apercevoir le mécanisme en détail, cf le protocole de Velayutham et al [12].

## B NetworkSimulator 2

NS est un simulateur de réseaux filaires et plus récemment de réseaux sans fils et ad-hoc. IL s'agit d'un projet *open source*, maintenu régulièrement, lancé par l'ISI (University of Southern California, USA). Le code est écrit en C++ et le langage de script OTcl sert d'interface pour décrire les simulations.

NetworkSimulator est reconnu par la communauté scientifique et son fonctionnement a été validé tout au long de son développement. Cependant le code est conséquent et effectuer des modifications peut vite tourner au cauchemar. Nous avons mis en place plusieurs méthodes (activation des RTS/CTS, changement de la période EIFS, variation des zones SR/TR...), et des scripts permettant d'extraire des graphiques à partir des fichiers de trace.

Il est à noter qu'un logiciel de visualisation des simulations est fourni, nommé NAM. Le site internet de NS-2: <http://www.isi.edu/nsnam/ns/>

## C Algorithme

Le code ne peut réellement être imprimé, car dissimulé dans de nombreux fichiers. Il a donc été écrit sous forme algorithmique. Les commentaires sont en anglais, puisque il s'agit de code intégré à NetworkSimulator.

It is a distributed algorithm, which goal is to adjust the contention window of each node regarding to their send-rate and their received-rate. Written in C++ language it is designed to be included into ns2-simulator.

*Nota:* the algorithm works sending pieces of information in CTS about neighbourhood and accounting. It could be easily rewritten sending information in ACK.

### *Initialisation*

```
memory_time = 1;           // seconds, period of data collect
expunge_time = memory_time + 0.; // seconds, period of silent-node recognition
cycle = 10;                // packets, number of packets send,received
                             // before recalling check_cw()
Fi = 1;                    // fairness index ( Wi over Wo )
precis = 10e5;             // a quotient to send Wo in integer type
                             // into CTS or ACK
plus_ = 1;                 // meter of packets sends
moins_ = 1;                // meter of packets received
C_ = 2;                    // percentage, error on Fi permitted
```

### *every transmissions*

```
if (Fi < 1) increment(moins_); // the fonction increment sets the variable to null
if (Fi > 1) increment(plus_);  // when cycle is reached

add(Wei, destination, packet_duration);
```

### *sending CTS*

```
add(CTS_packet, neighbourhood);
add(CTS_packet, sigma(Wo));
```

### *every receptions*

```
expunge(Wi, Wo); // a function to expunge the fifo data of packet
```

```

// accounting
add_neighbour(source); // add the source in the neighbour list

add(Wo, source, packet_duration); // update packet accounting

switch (packet_type) {
  case CTS: {
    if !(is_neighbour(destination)) add(Woo, source, RTS_duration);
    // the CTS packet is not for me and not for a node
    // that I heard => to exhib a three-flow transmission
    recover(source_neighbourhood, source_Wo);
  }
  case ACK: if !(is_neighbour(destination)) add(Woo, source, DATA_duration);
    // ditto, the ACK packet is not for me and not for
    // node that I heard (field DATA size in in ACK frame)
}

compute (Wo_ // with data received in CTSs, computes the global Wo
// avoids conflicts with personal Wo, ie takes account
// of my neighbourhood and the sending one's. That's
// why we send the sum and the neighbourhood

Fi = Wi / Wo;

if (Fi > (Fi +% C_)) {

  if ((plus_ == 0) && (Fi_evolution_i > 0.9))
    assign(cw_, cw_ * Wi_evolution * Fi_evolution);

  if (plus_ == 0) {
    plus_ = 1;
    remember(Fi, Wi);
  }
} else
if (Fi > (Fi -% C_)) {

  if ((moins_ == 0) && (Fi_evolution_i < (1.1)))
    assign(cw_, cw_ * Fi_evolution / Wo_evolution);

  if (moins_ == 0) {
    moins_ = 1;
    remember(Fi, Wo);
  }
} else {

  if (moins_ == 0) moins_ = 1;
  if (plus_ == 0) plus_ = 1;
}

if (Fi < 1) increment(moins_);
if (Fi > 1) increment(plus_);

```

## D Equipe d'accueil et remerciements

Ces travaux ont été effectués au sein du laboratoire CITI (Centre d'Innovations en Télécommunications & Intégration de services), unité mixte de recherche (Inria, Insa Lyon) qui se situe sur le campus de l'Insa de Lyon. Le CITI ne borne pas sa recherche sur les réseaux ad-hoc à l'étude la couche MAC. Bien au contraire, puisque les thèmes abordés vont du traitement du signal jusqu'aux problèmes de routage.

Ce stage s'est effectué sous la direction d'Isabelle Guérin-Lassous, chargée de recherche Inria et de ses deux étudiants en thèse, Dominique Dhoutaut et Claude Chaudet. Leurs dernières recherches portent sur *l'allocation équitable de bande passante*, des *protocoles intégrant de la qualité de service* pour les réseaux ad-hoc et les faiblesses du protocole MAC de 802.11. C'est dans cette dernière thématique qu'ils souhaitaient aborder les problèmes d'équité. Ma contribution fournit à l'équipe un point de vue global sur le problème et des pistes de recherche. Elle met surtout en lumière des problèmes méconnus. Enfin le code implémenté sous NS-2 permettra de simuler rapidement un algorithme de gestion de fenêtre de contention plus performant.

Je les remercie vivement pour l'attention qu'ils m'ont portée et leur accueil chaleureux.



Serveur du CITI: <http://citi.insa-lyon.fr>

## Table des figures

1	Cycle de transmission d'un paquet . . . . .	4
2	Problème de la diffusion radio . . . . .	5
3	Le problème des nœuds cachés . . . . .	6
4	Le cycle de transmission avec les RTS/CTS . . . . .	7
5	Une situation délicate . . . . .	8
6	Chronogramme associé au scénario de la figure 5 . . . . .	8
7	Équité par flux . . . . .	9
8	Scénario nœuds cachés & exposés : résultats . . . . .	11
9	Dessin synoptique des flux en contention avec $\mathcal{F}$ . . . . .	12
10	Dans l'ordre: exemple, flow contention graph $\mathcal{G}$ et ressource graph . . . . .	12
11	Les deux problèmes de l'EIFS . . . . .	14
12	Large EIFS: chronogramme – sans RTS/CTS . . . . .	14
13	Small EIFS: chronogramme – avec RTS/CTS . . . . .	15
14	EIFS: résultats des simulations . . . . .	15
15	Le scénario des trois paires . . . . .	16
16	Les trois paires: chronogramme & résultats . . . . .	16
17	Compétition entre les nœuds: aide au calcul de $W_o$ . . . . .	19
18	Scénario 1 – évolution de FI au cours du temps . . . . .	20
19	Scénario 1 – débit à gauche et évolution de CW à droite . . .	21
20	Scénario 2 – débits en regard de ceux expérimentés sous 802.11	21
21	Scénario 2 – évolution de CW à gauche et de FI à droite . . .	22