
Evaluation d'un protocole de régulation de débit dans les réseaux sans fil multisauts

Rémi Vannier* — Tahiry Razafindralambo** — Isabelle Guérin Lassous***

(*)INRIA / LIP

(**)CNRS/INRIA/Univ. Lille 1

(***)Univ. Lyon 1 / LIP

remi.vannier@ens-lyon.fr; tahiry.razafindralambo@inria.fr; isabelle.guerin-lassous@ens-lyon.fr

RÉSUMÉ. Le standard IEEE 802.11 est très souvent considéré comme la technologie sans fil sous-jacente dans les réseaux radio multisauts. Il est maintenant bien connu que l'utilisation de ce standard pose des problèmes d'efficacité et d'équité dans de tels réseaux. Ces problèmes proviennent en partie du protocole MAC de 802.11. Une approche possible est de proposer une alternative à ce protocole MAC afin de limiter ou d'éliminer ces problèmes. Néanmoins, il faudra très probablement attendre un certain temps avant de voir apparaître, sur le marché, des cartes sans fil adaptées aux réseaux multisauts. Une autre approche est de considérer que la technologie sous-jacente restera 802.11 pour un certain temps et que les solutions doivent se situer au-dessus de 802.11. Dans [VAN 08], nous avons proposé un algorithme d'allocation de bande passante distribué et dynamique qui reposait sur un modèle simple de partage du médium radio. Dans cet article, nous transformons cet algorithme en un protocole de régulation de débit pour les réseaux sans fil multisauts. Ce protocole, appelé *Profiterole*, est distribué, asynchrone, dynamique et nécessite peu d'informations sur le réseau. Puis, nous proposons une évaluation détaillée de *Profiterole* menée sous le simulateur de réseaux NS2. Enfin, nous le comparons à *DiffQ*, un nouveau protocole de contrôle de congestion dans les réseaux sans fil multisauts.

ABSTRACT. The IEEE 802.11 standard is often considered as the underlying wireless technology used in multihop wireless networks. The use of this standard in these networks raises some issues in terms of efficiency and fairness. These issues come mainly from the MAC protocol of 802.11. An approach is to propose an alternative to this MAC protocol in order to solve or decrease these problems. But it should probably take some time before new wireless network cards be based on these solutions. Another approach is to consider that IEEE 802.11 will be the wireless local technology for a while and that the solutions should be designed above. In [VAN 08], we describe a simple bandwidth allocation algorithm that is distributed and dynamic. This algorithm is based on a simple wireless medium model. In this article, we derive a throughput regulation protocol from this algorithm for multihop wireless networks. This protocol, called *Profiterole*, is distributed, asynchronous, dynamic and uses few control information. Then, we give a detailed evaluation of *Profiterole* thanks to NS2 simulation results. We also compare it with *DiffQ*, a new and recent congestion control algorithm for multihop wireless networks.

MOTS-CLÉS : réseaux sans fil multisauts, 802.11, contrôle de débit, équité

KEY WORDS: wireless multihop networks, 802.11, throughput regulation, fairness

1. Introduction

IEEE 802.11 est souvent considéré comme la technologie sans fil sous-jacente utilisée dans les réseaux sans fil multisauts. Il est maintenant bien connu que l'utilisation de 802.11 dans ce contexte pose un certain nombre de problèmes en termes d'équité et d'efficacité [CHA 05]. Différentes solutions ont été proposées pour tenter de réduire ou résoudre ces problèmes. Une approche possible est de proposer des alternatives au protocole MAC de 802.11 qui est en partie responsable de ces problèmes. Si ces solutions sont intéressantes, il est néanmoins peu probable qu'elles soient implantées rapidement dans les cartes sans fil vendues sur le marché. Une autre approche est de considérer que le protocole MAC de 802.11 tel qu'il a été initialement défini a encore une longue vie devant lui¹ et que les solutions aux problèmes soulevés précédemment doivent donc être mises au point au-dessus de 802.11.

Certaines solutions suggèrent de réguler le débit entrant au niveau de la couche MAC de 802.11. En régulant les débits entrants, on limite l'apparition de congestion et ainsi on réduit ou on élimine les problèmes d'efficacité et d'équité engendrés par la congestion. La régulation des débits entrants au niveau MAC peut se faire via une allocation des débits des flux. Pour réaliser une telle allocation, les solutions basées sur cette approche reposent sur un modèle de contention du médium radio qui cherche à capturer les dépendances entre les liens sans fil. La plupart de ces solutions (qui ne sont pas si nombreuses) se basent sur un graphe de contention des liens dans lequel il faut identifier les cliques maximales ou les ensembles indépendant maximaux pour calculer une allocation optimale des débits.

Dans [VAN 08], nous avons proposé un algorithme de partage équitable de bande passante pour les réseaux sans fil multisauts. Cet algorithme, comparé aux solutions existantes, repose sur un modèle de contention par nœuds simple. Il est de plus décentralisé et adaptatif, qualités essentielles pour de tels réseaux. Dans cet article, nous complétons l'idée initiale de [VAN 08] : i) nous montrons comment l'algorithme d'allocation de débits peut être adapté en un protocole réseau de régulation de débits ; ii) nous proposons une évaluation de ce protocole, appelé *Profi terole*, réalisée par simulations avec le simulateur NS2, puis iii) nous comparons ce protocole avec un protocole récent de régulation de débits, appelé *DiffQ* et présentant de bonnes performances [WAR 09].

Cet article est organisé de la façon suivante : la section 2 donne un rapide état de l'art sur les solutions de régulation de débits dans les réseaux sans fil multisauts ; la section 3 montre comment l'algorithme de [VAN 08] est transformé en un protocole réseau ; enfin la section 4 décrit les résultats de performance de notre protocole et les compare avec ceux obtenus pour *DiffQ*.

2. État de l'art

Les algorithmes d'allocation de bande passante fonctionnent en général en deux étapes. La première consiste à calculer les dépendances entre les liens sans fil selon un modèle de contention du médium radio. La deuxième étape est l'allocation des débits en fonction des dépendances calculées

1. 802.11e est un des rares amendements qui modifie le protocole MAC défini initialement dans 802.11 et à notre connaissance, il n'existe pas de cartes sans fil 802.11e sur le marché.

à l'étape précédente. Ces tâches sont d'autant plus difficiles qu'elles doivent être réalisées dans un contexte sans fil, multisauf et mobile.

Dans [GUP 00], les auteurs introduisent deux modèles de contention : le *modèle protocolaire* et le *modèle physique*. Dans le modèle protocolaire, un récepteur peut recevoir avec succès les paquets envoyés par un émetteur si les deux nœuds sont à portée de communication et si aucun autre nœud n'est en train de transmettre dans la zone d'interférence du récepteur. Il est possible d'adapter ce modèle au protocole MAC de 802.11 en assurant de plus qu'aucun autre nœud n'est en train de transmettre dans la zone de détection de porteuse et dans la zone d'interférence (pour une réception correcte des ACKs) de l'émetteur. Dans le modèle physique, une transmission se fait correctement si le rapport signal-sur-bruit au niveau du récepteur est suffisamment important (i.e. supérieur à un certain seuil). Ces deux modèles sont souvent considérés comme les modèles de base à partir desquels les modèles de partage du médium radio sont construits afin d'exprimer les dépendances entre les flux du réseau.

Dans [LUO 00], les auteurs construisent un *graphe de contention des liens sans fil*. Dans un tel graphe, les sommets correspondent aux liens sans fil du réseau et il y a une arête entre deux sommets dans le graphe si deux flux passant par les deux liens sans fil correspondants dans le réseau sont en contention. La contention peut être modélisée par un des deux modèles décrits précédemment. A partir de ce graphe, différentes méthodes sont proposées pour écrire les contraintes entre les débits des flux.

Toujours dans [LUO 00], les auteurs identifient les ensembles indépendants maximaux dans le graphe de contention des liens sans fil. Ces ensembles donnent lieu à des conditions nécessaires et suffisantes qui assurent qu'il existe un ordonnancement des communications dans le réseau tel que deux nœuds voisins dans le graphe (et donc deux liens sans fil dans le réseau) ne sont pas actifs en même temps. Un des problèmes avec cette approche est que le calcul des ensembles indépendants maximaux est coûteux et souvent difficile à réaliser de manière distribuée. De plus, dans ce travail, le graphe de contention repose sur le modèle protocolaire dans lequel la zone de détection de porteuse et la zone d'interférence correspondent à la zone de communication.

Une deuxième approche consiste à considérer des contraintes sur les cliques maximales dans le graphe de contention des liens. Dans une clique maximale, les liens sans fil associés sont en compétition pour l'accès au médium radio. L'avantage des cliques maximales, comparées aux ensembles indépendants maximaux, est que ce sont des structures plus locales qui peuvent être plus facilement déterminées localement. L'inconvénient est qu'elles ne donnent lieu qu'à des contraintes nécessaires, ce qui implique qu'un ordonnancement sur l'activation des liens sans fil peut ne pas exister bien que les contraintes sont vérifiées [JAI 03].

Le travail de [XUE 06] est un des tout premiers à proposer un protocole d'allocation équitable sur les débits des flux à partir des cliques maximales. Les auteurs proposent une construction des cliques décentralisée sur le graphe de contention des liens, lui-même basé sur le modèle protocolaire où la zone d'interférence est considérée comme étant soit les liens à un saut soit les liens à deux sauts. A partir du calcul des cliques, les auteurs mettent au point une allocation distribuée des débits faisant appel à l'optimisation lagrangienne.

Dans [GUP 07], les auteurs utilisent les *contraintes par ligne* obtenues via la matrice d'incidence du graphe de contention : pour chaque lien, la somme du débit de ce lien et des débits des liens

voisins (dans le graphe de contention) ne doivent pas dépasser la capacité du médium radio. De telles contraintes sont suffisantes mais non nécessaires, ce qui peut donner lieu à une allocation éloignée de l'optimal. Néanmoins, ces contraintes ont l'avantage d'être beaucoup plus simples à calculer que les cliques maximales.

Dans [VAN 08], nous proposons d'utiliser un graphe de contention plus simple que le graphe de [LUO 00] sur lequel les approches précédentes se basent. Dans notre modèle, le graphe de contention se déduit directement du réseau de communication : les nœuds du graphe correspondent aux nœuds du réseau et il y a un lien entre deux nœuds dans le graphe de contention si les deux nœuds sont à au plus deux sauts dans le réseau. Ce modèle par nœud provient de résultats expérimentaux qui montrent que la distance de détection de porteuse correspond à environ deux fois la distance de communication obtenue avec des débits physiques de base (comme 2 Mb/s par exemple). A partir du graphe de contention, nous écrivons les contraintes par ligne qui expriment le partage du médium radio à l'émission, puis dérivons une allocation des débits via l'optimisation lagrangienne. Les premiers résultats présentés dans [VAN 08] montrent que les résultats de cette approche sont aussi bons que ceux de [XUE 06] en termes de débit total et de convergence, tout en ayant une complexité moindre.

Tout récemment, le protocole DiffQ a été proposé dans [WAR 09]. Le but de ce protocole n'est pas de répondre à une allocation de débit spécifique, mais plutôt de réaliser une régulation du débit lorsqu'une congestion apparaît, comme le fait TCP. Le point intéressant de ce protocole est que la détection de congestion se fait localement et que le contrôle de congestion se fait en tout point du chemin sans attendre la perte de paquets, comme le fait TCP par exemple. Le point faible de ce protocole est que le contrôle de congestion nécessite de modifier la couche MAC du standard 802.11. Nous avons néanmoins décidé de retenir cette solution pour comparaison car elle présente actuellement de très bonnes performances par rapport aux solutions existantes.

3. Profiterole : un protocole de régulation de débits

Dans cette section, nous décrivons rapidement l'algorithme d'allocation de débit que nous avons proposé dans [VAN 08], puis nous montrons comment cet algorithme peut être dérivé en un protocole réseau, appelé Profiterole : PROportionnal FaIr raTE contROL protocol.

3.1. L'algorithme

Le problème de maximisation globale des débits des flux peut s'écrire sous forme d'un problème d'optimisation avec contraintes linéaires :

$$\begin{aligned} \text{MAXIMISER :} & \quad \sum_{f \in \Phi} \log \phi_f \\ \text{SOUS LES CONTRAINTES :} & \quad \forall n \in N \quad \sum_{i \in V_2(n)} x_i \leq C \end{aligned}$$

où :

– C est la capacité du médium radio.

- $V_2(n)$ est le voisinage à deux sauts du nœud n .
- x_i est le débit d'émission (tous flux confondus) alloué au nœud i .
- ϕ_f est le débit du flux f .
- La fonction objectif, $\sum_{f \in \Phi} \log \phi_f$, est un compromis entre maximisation de l'utilisation du medium radio et équité entre flux. Elle atteint son maximum pour l'allocation *proportionnelle-équitable*.

Le problème d'optimisation dual de ce problème peut être résolu de manière décentralisée par des méthodes d'optimisation Lagrangienne. Sans rentrer dans les justifications théoriques (voir pour référence [KEL 98]), nous donnons les grandes lignes de l'algorithme. A chaque nœud n est associé un *tarif* λ_n dont il est responsable. Ce *tarif* représente le degré de saturation de ce nœud et indique à un flux le *prix* à payer pour utiliser la bande passante de ce nœud, que ce soit parce que ce flux est émis par le nœud n ou par des nœuds à deux sauts de ce nœud n . A chaque pas s de l'algorithme :

- Les nœuds calculent de manière distribuée et pour chaque flux f le prix total que celui-ci devra payer pour emprunter cette route. Ce *prix* sera noté $\Pi_f^{(s)}$. Si r_{nf} est le nombre de nœuds émetteurs diffusant le flux f dans le voisinage à deux sauts de n , alors :

$$\Pi_f^{(s)} = \sum_{n \in N} r_{nf} \times \lambda_n^{(s)} \quad (1)$$

- Le *prix* d'un flux est alors diffusé à l'ensemble des nœuds concernés par ce flux, i.e. les nœuds se trouvant à deux sauts de ce flux.

– Les nœuds réévaluent alors leur tarif en fonction de ces prix. Un nœud beaucoup utilisé va augmenter son tarif, alors qu'un nœud non saturé verra son tarif descendre à 0. A chaque pas, chaque nœud recalcule son *tarif* selon la formule :

$$\lambda_n^{(s+1)} = \lambda_n^{(s)} + \sigma \times \left(C - \sum_{f \in \Phi} \frac{r_{nf}}{\Pi_f^{(s)}} \right) \quad (2)$$

On peut montrer qu'après quelques itérations de l'algorithme, la valeur unique vers laquelle convergent les prix des flux est l'inverse du débit auquel ils peuvent émettre.

3.2. Le protocole *Profiterole*

Transformer un algorithme en protocole pose un certain nombre de challenges qui n'apparaissent pas dans l'algorithme, comme la gestion de l'asynchronisme, la perte de paquets de contrôle ou le surcoût induit pas 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 1 saut (incluant le tarif du nœud qui envoie le message Hello). Ainsi chaque nœud connaît les tarifs de tous ses voisins à 2 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 destinataire 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 1. 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 2-voisinage, et adapte son tarif en conséquence selon la formule 2.

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 seconde², et aux *price messages* eux aussi envoyés sur une base de 1 paquet par flux, par saut, et par seconde.

4. Evaluation de performance

Dans cette partie nous donnons les résultats de simulation dans lesquels nous comparons le protocole *Profiterole* au protocole *DiffQ*. Nous utilisons le simulateur ns-2 [VIN 00] dans sa version 2.33 avec l'extension CMU pour les réseaux ad hoc. Les autres paramètres de simulation sont présentés dans le Tableau 1. Certains paramètres de *DiffQ* n'étant pas donnés dans [WAR 09], nous les avons choisis arbitrairement en prenant en compte les recommandations des auteurs. 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	[WAR 09]
		AIMD : $\alpha = 2/3, \beta = 1kbps$
		Queue_Thresh = 10
Profiterole	[VAN 08]	
Trafic	CBR / UDP	Capacité max
	FTP / TCP	Capacité max
	Départ des flux	Aléatoire

Tableau 1. 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.

2. Rappelons 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.

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 1(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 4.1 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 1(b)), nous voulons faire apparaître le problème des interférences intra-flux. Les résultats présentés dans la section 4.2 montrent que DiffQ et Profiterole obtiennent de bons résultats. Dans la troisième partie donnée en section 4.3, 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.

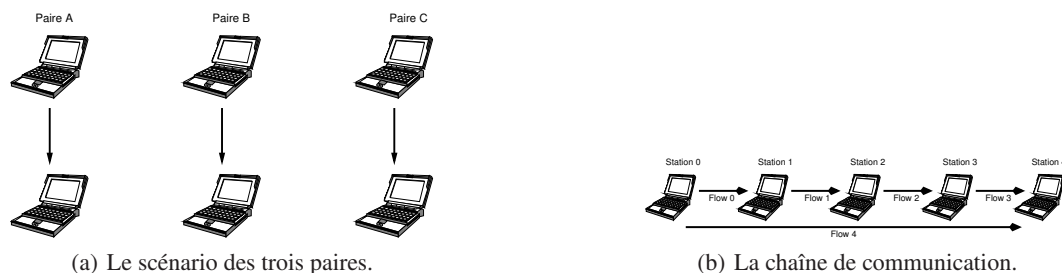


Figure 1. Scénarios étudiés.

4.1. Trois paires

Le scénario des trois paires Fig. 1(a) a été étudié dans [CHA 05]. 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 2(a)). Pour éviter cette famine, un partage de la bande passante est nécessaire.

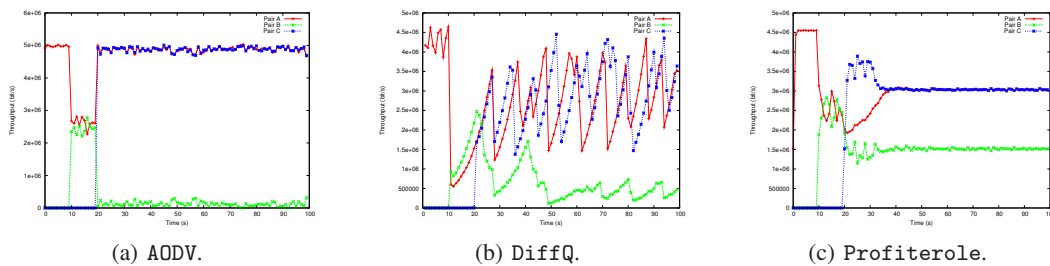


Figure 2. Evolution du débit en fonction du temps dans le scénario des trois paires.

La figure 2 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 2(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.

4.2. Chaîne de communication

Dans le scénario de la chaîne de communication, présenté dans la figure 1(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.

La figure 3 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 3(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

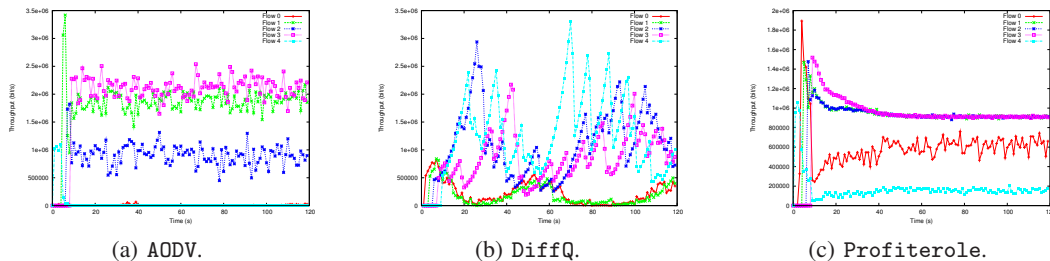


Figure 3. Evolution du débit en fonction du temps dans le scénario de la chaîne de communication.

que contrairement à *Profiterole* et *AODV*, le flux le plus avantageé 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 [BHA 94]).

4.3. 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.

4.3.1. Flux UDP

Nous choisissons aléatoirement 5 sources et 5 destinations avec des flux UDP/CBR. La figure 4 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 4 que, comme pour la chaîne de communication, *Profiterole* permet d'avoir plus d'équité au niveau des débits. En effet, nous pouvons voir 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

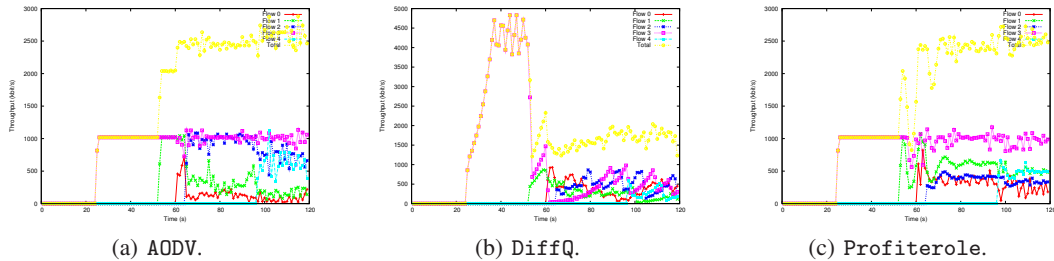


Figure 4. Evolution du débit en fonction du temps sur un scénario aléatoire (30 nœuds, 5 flux, udp).

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 2.

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

Tableau 2. Résumé des débits obtenus par simulation. Scénario aléatoire (30 nœuds, 5 flux, udp).

4.3.2. Flux TCP

Dans cette section, nous choisissons aléatoirement 2 sources et 2 destinations avec des flux TCP et un trafic FTP. La figure 5 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.

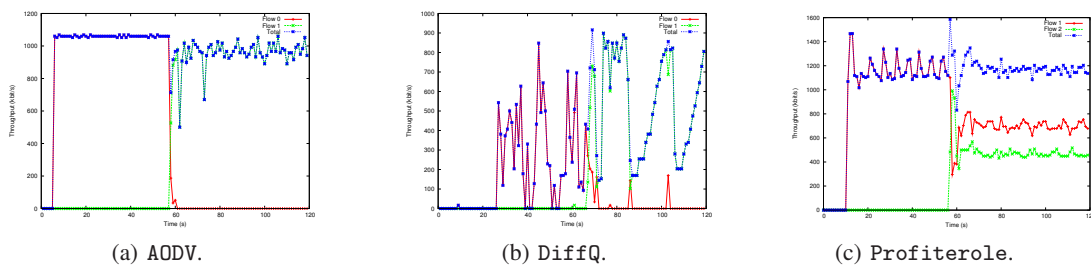


Figure 5. Evolution du débit en fonction du temps sur un scénario aléatoire (30 nœuds, 2 flux, tcp)

Les résultats de la figure 5 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 *DiffQ*³. 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 5 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 3.

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

Tableau 3. Résumé des débit obtenus par simulations. Scénario aléatoire (30 nœuds, 2 flux, tcp).

5. Conclusion

Cet article s'intéresse au problème de la régulation des débits dans les réseaux sans fil multisauts basés sur la technologie IEEE 802.11. Dans cet article, nous proposons un protocole de régulation des débits appelé *Profiterole*. A partir d'information sur le 2-voisinage et sur les routes empruntées par les flux, *Profiterole* réalise une régulation équitable des débits des flux sans pour autant sacrifier l'utilisation globale du médium radio. Les résultats obtenus sur NS2 montrent un bon comportement de *Profiterole* sur différents scénarios et ce aussi bien avec des flux UDP qu'avec des flux TCP. Enfin, nous avons comparé *Profiterole* au protocole *DiffQ*, nouvelle solution de contrôle de congestion pour les réseaux sans fil multisauts 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*.

Dans le futur, nous envisageons d'implanter *Profiterole* sur machines réelles afin d'évaluer ses performances dans un contexte sans fil réel. L'avantage de *Profiterole* est qu'il ne nécessite aucune modification de 802.11. Nous aimerions aussi adapter *Profiterole* à certains protocoles de qualité de service. Le but est d'obtenir un protocole capable d'assurer des garanties aux flux contraints tout en réalisant un partage équitable et efficace du médium restant pour les flux élastiques. *Profiterole* pourrait réaliser ce partage pour ces derniers.

³. les débits sont calculés à partir de 100s.

6. Bibliographie

- [BHA 94] BHARGHAVAN V., DEMERS A., SHENKER S., ZHANG L., « MACAW : a media access protocol for wireless LAN's », *ACM SIGCOMM Computer Communication Review*, London, United Kingdom, 1994, p. 212-225.
- [CHA 05] CHAUDET C., DHOUTAUT D., GUÉRIN-LASSOUS I., « Performance Issues with IEEE 802.11 in Ad Hoc Networking », *IEEE Communication Magazine*, vol. 43, n° 7, 2005, p. 110-116.
- [GUP 00] GUPTA P., KUMAR P. R., « The Capacity of Wireless Networks », *IEEE Transactions on Information Theory*, vol. 34, n° 5, 2000, p. 910-917.
- [GUP 07] GUPTA R., MUSACCHIO J., WALRAND J., « Sufficient Rate Constraints for QoS Flows in Ad-Hoc Networks », *Ad Hoc Networks*, vol. 5, n° 4, 2007, p. 429-443, Elsevier Science Publishers B. V.
- [JAI 03] JAIN K., PADHYE J., PADMANABHAN V. N., QIU L., « Impact of Interferences on Multi-hop Wireless Network Performance », *ACM Mobicom*, 2003.
- [KEL 98] KELLY F., MAULLOO A., TAN D., « Rate control in communication networks : shadow prices, proportional fairness and stability », *Journal of the Operational Research Society*, vol. 49, 1998.
- [LUO 00] LUO H., LU S., BHARGHAVAN V., « A New Model for Packet Scheduling in Multihop Wireless Networks », *ACM Mobicom*, 2000, p. 76-86.
- [VAN 08] VANNIER R., LASSOUS I. G., « Partage équitable de la bande passante dans les réseaux ad hoc », *CFIP*, Les Arcs, France, March 2008.
- [VIN 00] VINT PROJECT TEAM, « The network simulator – ns-2 », VINT Project Team, Available at <http://www.isi.edu/nsnam/ns/>, November 2000.
- [WAR 09] WARRIER A., JANAKIRAMAN S., RHEE I., « DiffQ : Practical Differential Backlog Congestion Control for Wireless Networks », *Infocom*, 2009.
- [XUE 06] XUE Y., LI B., NAHRSTEDT K., « Optimal Resource Allocation in Wireless Ad Hoc Networks : A Price-based Approach », *IEEE Transactions on Mobile Computing*, vol. 5, n° 4, 2006.