

Modélisation et évaluation de performances des stratégies de backoff

Tahiry Razafindralambo, Fabrice Valois

► **To cite this version:**

Tahiry Razafindralambo, Fabrice Valois. Modélisation et évaluation de performances des stratégies de backoff. Colloque Francophone sur l'Ingénierie des Protocoles - CFIP 2006, Oct 2006, Tozeur/Tunisia, Hermès, 12 p., 2006, Session 6 : qualité de service et réseaux sans fil. <inria-00111903>

HAL Id: inria-00111903

<https://hal.inria.fr/inria-00111903>

Submitted on 20 Nov 2006

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.

Modélisation et évaluation de performances des stratégies de *backoff*

Tahiry RAZAFINDRALAMBO¹ — Fabrice VALOIS

Laboratoire CITI - INRIA ARES - INSA Lyon
21 avenue Jean Capelle - 69621 Villeurbanne Cedex - FRANCE
e-mail : tahiry.razafindralambo@insa-lyon.fr, fabrice.valois@insa-lyon.fr

RÉSUMÉ. La couche MAC et, en particulier, le mécanisme d'évitement de collisions (*backoff*) des réseaux radio est un mécanisme clef dans l'efficacité de ces réseaux. Si celui-ci est trop agressif, l'équité sera mauvaise alors qu'un mécanisme plus relâché offrira des performances plus faibles. Cela est vrai tant pour les réseaux en mode infrastructure que pour les réseaux multi-sauts. Nous nous intéressons ici à la modélisation et à l'évaluation de performances de différentes stratégies de *backoff* dans ces deux contextes. Nous mettons en avant la nécessité de recourir à des stratégies de *backoff* différentes entre les réseaux à 1 saut et multi-sauts à cause du compromis efficacité-équité. Les mesures de performances traduisent l'efficacité des mécanismes mais aussi leur équité à travers la proposition d'une nouvelle métrique.

ABSTRACT. In the widely used 802.11 standard, stochastic analysis allows *backoff* algorithms performance evaluation. Many papers show that same protocols can have different behaviours depending on the topology. In this paper, we propose a stochastic analysis of 802.11, and different *backoff* algorithms. We focus our work on both single hop and multi-hop networks. Efficiency and fairness are the performance evaluation metrics we use in our analysis. Results show that different *backoff* can show the same behaviour in single hop networks but can behave differently in multi hop context. We also show the tradeoff between efficiency and fairness.

MOTS-CLÉS : Réseaux sans fil, modélisation, algorithme de *backoff*, performances, équité

KEYWORDS: Wireless LANs, 802.11, *backoff* algorithms, performance evaluation, fairness

1. Financé à travers le CRE-46128746 France Telecom R&D.

1. Introduction

La rapidité de déploiement, le coût bon marché et les performances des réseaux 802.11 ont permis leur déploiement massif en mode infrastructure. La disponibilité de cette interface radio et sa flexibilité ont permis son utilisation pour expérimenter des réseaux multi-sauts où chaque noeud collabore au réseau : les performances mesurées sont alors bien en deçà des attentes. La raison vient du protocole d'accès au médium et du mécanisme d'évitement de collisions (*backoff*) qui ont été conçus pour fonctionner en mode infrastructure uniquement.

Il est donc nécessaire de proposer de nouveaux protocoles MAC pour réseaux ad hoc. Plusieurs propositions ont vu le jour comme : émission probabiliste ou modification du comportement du *backoff*, etc. Mais avant de proposer de telles solutions, il nous semble important d'analyser en profondeur les algorithmes de *backoff* déjà proposés, de comprendre et de justifier en quoi les contraintes d'un réseau radio en mode infrastructure sont fondamentalement différentes de celles d'un réseau multi-sauts. Nous proposons donc de modéliser les réseaux 802.11 en considérant ces deux topologies. Nous utilisons ici une algèbre de processus stochastique dédiée à l'évaluation de performances : PEPA [HIL 94] (*Performance Evaluation Process Algebra*). La motivation d'utiliser PEPA plutôt que les outils classiques de modélisation stochastique est double : *i*) les réseaux modélisés conduisent à un large espace d'états et les techniques de réduction de cet espace proposées dans PEPA sont bien plus performantes que les traditionnelles techniques d'agrégation stochastique *ii*) l'approche compositionnelle des algèbres de processus conduit à des modèles apparaissant comme génériques : il est donc *aisé* d'étendre, de modifier un modèle pour étudier une nouvelle topologie réseaux ou un nouvel algorithme de *backoff*, tout en exploitant le travail fait auparavant. Nous proposons d'étudier plusieurs stratégies de *backoff* dans les environnements suivants : réseau 802.11 en mode infrastructure pour n noeuds, réseau multi-sauts avec la topologie des noeuds cachés et celle des 3 paires [CHA 04].

Le présent article est ainsi structuré : nous débuterons par une introduction à l'outil de modélisation utilisé suivi d'un état de l'art sur l'évaluation de performances de 802.11. Ces préambules sont les acquis nécessaires au travail présenté par la suite. La section 4 présentera les principales stratégies de *backoff* et détaillera la modélisation d'un réseau sans fil. La section 5 sera consacrée aux résultats d'évaluation de performances de ces *backoff* sur différentes topologies. De traditionnels résultats de performances (débit, taux de collisions, etc.) seront complétés par une mesure de l'équité de ces protocoles. Nous concluons ce travail en précisant les prochaines pistes de recherche dans la section 6.

2. Introduction à l'algèbre de processus stochastique PEPA

PEPA (*Performance Evaluation Process Algebra*) [HIL 94] étend la notion classique des algèbres de processus en associant une variable aléatoire à chaque action. Cette variable aléatoire modélise une durée et suit une loi exponentielle conduisant à une relation évidente entre un modèle décrit sous forme d'algèbre de processus sto-

chastique et une chaîne de Markov à temps continu. A partir de ce processus markovien, il est possible de dériver des critères de performances.

Le formalisme de PEPA : Les modèles PEPA sont décrits comme des interactions de plusieurs *composantes*. Chaque composante peut suivre le comportement de une ou plusieurs *actions* : une action $a \in \mathcal{Act}$ est décrite par le couple (α, r) où $\alpha \in \mathcal{Act}$ est le *type* de l'action et $r \in \mathbb{R}^+$ est le paramètre de la loi exponentielle liée à cette action. Un ensemble d'opérateurs est proposé dans PEPA pour construire des modèles complexes à partir de composantes exprimées simplement. Ce sont les opérateurs classiques des algèbres de processus : *Prefix*, *Choice*, *Parallel composition*, *Abstraction*. Dans la suite, nous décrivons les opérateurs que nous utilisons. Pour plus de renseignements sur le formalisme PEPA, nous vous invitons à consulter [HIL 94].

- *Prefix* : Une composante peut avoir un comportement séquentiel dans lequel elle suit une succession d'actions avant de suivre le comportement d'une autre composante. Dans ce cas, l'opérateur *prefix*, noté “.”, est utilisé pour désigner la première action, e.g. $(\alpha, r).P$ se comportera d'abord comme une activité de type α avec une durée moyenne de $1/r$ puis aura le comportement de la composante P . Dans certains cas, le taux de l'activité n'est pas spécifié, car non connu lors de l'écriture du modèle. Ce taux sera alors acquis par synchronisation avec une autre composante possédant le taux. Dans ce cas, le taux d'activité est spécifié avec un symbole particulier : \top et on parle d'action passive.

- *Choice* : Cet opérateur modélise le choix entre deux comportements, e.g. $(\alpha, r).P + (\beta, s).Q$. La nature continue des distributions de probabilités nous garantit que ces deux actions ne peuvent pas se produire simultanément. Le comportement final sera celui de la première activité à se terminer.

- *Parallel composition* : Cet opérateur est utilisé lorsque deux composantes doivent collaborer à travers certaines actions. Il permet donc la synchronisation entre deux composantes. Par exemple, le système $P \underset{L}{\bowtie} Q$ décrit deux composantes P et Q devant collaborer dans les actions définies dans l'ensemble de synchronisation L . Les actions non spécifiées dans L demeurent indépendantes et se déroulent dans P et Q sans être en concurrence. Les actions spécifiées dans L doivent nécessairement se dérouler simultanément dans P et Q pour modéliser la synchronisation. L'action résultant de cette synchronisation conserve le même *type* que celui défini dans P et Q mais le taux résultant est le plus petit taux d'activité des deux composantes. Cela implique que le taux d'une action passive sera le taux de l'action avec laquelle elle se synchronise.

Processus markovien : Dans un modèle PEPA, lorsqu'une composante P suit une activité (α, r) puis se comporte comme la composante P' , on dit que P' est dérivée de P . Pour chaque composante PEPA P , il est possible de construire récursivement un ensemble de dérivation (noté $ds(P)$). À partir de cet ensemble de dérivation, nous pouvons construire le *graphe de dérivation*. Ce graphe de dérivation est un graphe orienté tel que l'ensemble des nœuds est $ds(P)$ et un arc entre deux nœuds représente la possibilité d'une transition entre les deux composantes associées.

Grâce à l'utilisation de variables aléatoires suivant des distributions exponentielles, un modèle PEPA conduit à une chaîne de Markov à temps continu. La construction de

cette chaîne est basée sur le graphe de dérivation du modèle. Un état est associé à un noeud du graphe et les transitions entre les états de la chaîne sont les arcs du graphe.

Résoudre le processus markovien : PEPA Workbench [S.G 94] est un outil capable de résoudre numériquement les modèles PEPA en générant la distribution stationnaire du système. L'outil PRISM [M.K 01] propose également une interface permettant de traiter les modèles PEPA et de calculer un état stationnaire du processus markovien sous-jacent.

3. État de l'art

[BIA 00] propose de modéliser la stratégie de *backoff* de 802.11 sous forme de chaîne de Markov dans le cas de réseaux d'infrastructure. Un modèle du *Binary Exponential Backoff* est donc proposé permettant de calculer les principaux paramètres de performances en fonction du nombre de stations et du taux de collisions. Plusieurs travaux étudiant les algorithmes de *backoff* sont dérivés de ce modèle comme [CHA 05] qui a proposé DIDD (Double Increase Double Decrease). Avec ce modèle, s'il est assez aisé d'étudier plusieurs algorithmes de *backoff*, il est en revanche plus laborieux d'étudier plusieurs topologies et d'étendre la modélisation au cas des réseaux multi-sauts.

Dans ce contexte, [CHA 04] propose l'étude du scénario des 3 paires à travers une modélisation analytique. Les auteurs mettent en avant une étude en termes de performances (efficacité, ...) et en termes d'équité inhérent à ce scénario. En revanche, étendre ce modèle à d'autres topologies est complexe à cause de la dépendance système-modèle. [LI 04] propose l'étude des stations cachées, scénario classique dans le contexte multi-sauts, en s'appuyant sur les chaînes de Markov. Les auteurs soulignent le problème d'équité à court terme. L'étude couvre aussi bien le mécanisme d'accès classique que le mécanisme RTS/CTS. La réutilisation de ce modèle pour d'autres scénarios multi-sauts est également complexe.

A travers tous ces travaux, il faut souligner qu'aucun ne permet d'étudier le comportement d'un réseau 802.11 dans lequel chaque nœud utiliserait un algorithme de *backoff* différent. Il est important de souligner que le modèle proposé ici permet de le faire *aisément*. De plus l'utilisation de notre modèle sur des scénarios différents n'impose pas la reconstruction entière d'un modèle. Le travail proposé ici peut donc être perçu sous deux angles : 1) un travail de modélisation générique pour étudier les réseaux sans fil et 2) une étude détaillée du comportement de plusieurs algorithmes de *backoff* dans le cas des réseaux d'infrastructure et multi-sauts.

4. Modélisation

Un réseau sans fil est décrit par trois entités : le modèle de nœud décrivant l'accès au médium (ici : *Carrier sense*), le mécanisme de *backoff* et la description de la topologie décrivant les interactions entre les nœuds via le partage du médium. Nous associons une composante PEPA à chacune de ces entités et définissons des ensembles

de synchronisation modélisant la collaboration entre chaque composante. Cette section décrit ces composantes et les algorithmes de *backoff* que nous allons étudier.

4.1. Modélisation des algorithmes de backoff

Les algorithmes de *backoff* seront modélisés avec la composante BO_{i_j} où i est l'identité du noeud transmetteur et j une étape du *backoff* classiquement associée à une taille de fenêtre de contention. Afin d'apparaître le plus générique possible, chaque modèle est modélisé avec les mêmes ensembles de synchronisation. Il est donc aisé d'étudier un algorithme de *backoff* puis un autre sans redévelopper tout un modèle.

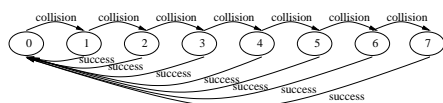


Figure 1. BEB

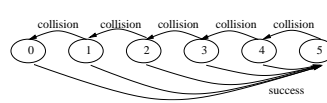


Figure 2. BEB inversé

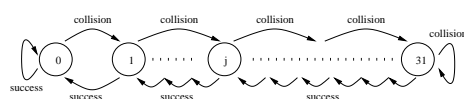


Figure 3. MILD

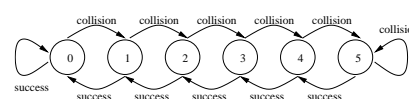


Figure 4. DIDD

L'algorithme de *backoff* classiquement utilisé dans 802.11 est le **Binary Exponential Backoff (BEB)**. La figure 1¹ modélise le comportement de BEB et le tableau 1 indique son modèle PEPA. Le principe de BEB est le suivant : à chaque collision, la taille de fenêtre est multipliée par deux sauf après la 8^{ième} collision consécutive où elle est réinitialisée. Après une transmission réussie, la fenêtre de contention reprend sa taille initiale. Pour modéliser le *backoff* nous utilisons n composantes ($n = 8$ pour BEB) et 3 activités ($succ_i, coll_i, db_i$) qui sont synchronisées avec le noeud i . $succ_i$ (resp. $coll_i$) est utilisé pour faire évoluer l'état du *backoff* en cas de transmission réussie (resp. échouée). db_i est utilisé pour modéliser le tirage et la décrémentation du *backoff*. Le taux f_j de l'activité db_i vaut : $f_j = 2 / ((2^j \times CW_{min} - 1) \times 20)^2$, où 20 représente la durée d'un slot, et la multiplication par 2 est utilisée pour représenter la valeur moyenne du tirage aléatoire dans la fenêtre de contention. Cela revient à modéliser le tirage aléatoire du *backoff* par un comportement moyen distribué suivant une loi exponentielle de ayant pour moyenne le temps moyen associé à la fenêtre de *backoff*. Notons que les temps d'attente (DIFS, SIFS, etc.) sont également modélisés par une loi exponentielle de moyenne la durée de cette durée xIFS. Les performances mesurées en section 5 valident cette hypothèse.

1. Dans cette figure, chaque état représente le numéro i de l'étape du *backoff* et donc de la fenêtre de contention
 2. Pour BEB, $f_6 = f_7 = f_5$.

<i>BEB</i>	$BO_{i,j}$	$\stackrel{def}{=} (db_i, f_j).BO_{i,j} + (succ_i, \top).BO_{i,0} + (coll_i, \top).BO_{i,(j+1)}$
<i>BEB inv</i>	$BO_{i,j}$	$\stackrel{def}{=} (db_i, f_{(7-j)}).BO_{i,j} + (succ_i, \top).BO_{i,7} + (coll_i, \top).BO_{i,(j-1)}$
<i>DIDD</i>	$BO_{i,j}$	$\stackrel{def}{=} (db_i, f_j).BO_{i,j} + (succ_i, \top).BO_{i,(j-1)} + (coll_i, \top).BO_{i,(j+1)}$
<i>MILD</i>	$BO_{i,j}$	$\stackrel{def}{=} (db_i, f_j).BO_{i,j} + (succ_i, \top).BO_{i,(j-1)} + (coll_i, \top).BO_{i,(2 \times j + 1)}$

Tableau 1. *Modèle PEPA des algorithmes de backoff*

A partir de ce modèle de BEB, nous pouvons définir le modèle de **BEB inversé** (voir figure 2). La fenêtre initiale correspond à la fenêtre maximale. A chaque collision, la taille de la fenêtre est divisée par 2. En revanche, dès que la transmission est acquittée, la fenêtre est ré-initialisée à sa taille maximale (voir tableau 1).

Le principe du *backoff Double Increase Double Decrease* [CHA 05] (DIDD) est le suivant : dès qu'une transmission collisionne la taille de la fenêtre de contention du nœud associé est multipliée par 2 et lorsqu'une transmission est réussie alors la taille de la fenêtre de contention est divisée par 2. La figure 4 illustre ce fonctionnement. Le modèle PEPA est présenté dans le tableau 1.

Dans *Multiplicative Increase Linear Decrease* (MILD) [BHA 94], dès qu'une transmission collisionne la taille de la fenêtre de contention du nœud associé est multiplié par 2 et lorsqu'une transmission est réussie la fenêtre de contention prend la valeur $CW_{suiv} = CW_{prec} - 32$. La figure 3 illustre le fonctionnement de MILD. Et le modèle PEPA est représenté dans le tableau 1. Ici $f_j = 2 / (((j + 1) \times CW_{min} - 1) \times 20)$.

4.2. Modélisation des réseaux d'infrastructure

Nous considérons au départ un scénario simple dans lequel 2 nœuds partagent le médium pour transmettre des informations à leur récepteur. Tous les nœuds sont à portée radio. Le modèle doit donc faire apparaître l'accès au canal radio.

Modèle du nœud : En mode infrastructure, tous les nœuds sont dans le même voisinage radio. Comme chaque nœud utilise le même protocole d'accès au médium (CSMA), ils ont tous le même comportement modélisé par la même composante : $E_{i,x}$ $i \in \{0,1\}$ où i est le nœud i . (voir tableau 2).

Dix actions sont utilisées pour modéliser le mécanisme d'accès au médium CSMA. L'action (*free*, μ_{trans}) sert à tester si le canal radio est libre ou non. Si le canal est libre cette action sert à synchroniser la station avec le canal radio. Si le médium est occupé, le nœud retourne à son état initial grâce à l'action (*occ*, μ_{dada}). L'action (*difs*, μ_{difs}) et l'action (db_i, \top), représentent l'écoulement des temps DIFS et du *backoff*. (db_i, \top) est une synchronisation avec l'algorithme de *backoff*. L'action (*send*, μ_{trans}) représente la capture du médium. Ici cette capture est représentée par un temps très court. Le temps de transmission s'écoule au niveau de la composante représentant le médium. Les actions (*ack_i*, \top) et (*collision*, \top) sont des synchronisations venant du médium spécifiant si la transmission s'est bien déroulée ou non. Les

E_{i_000}	$\stackrel{def}{=} (free, \mu_trans).E_{i_001};$	Med_{00_00}	$\stackrel{def}{=} (free, \top).Med_{00_00}$
E_{i_001}	$\stackrel{def}{=} (difs, \mu_difs).E_{i_002};$		$+ (send, \top).Med_{00_01};$
E_{i_002}	$\stackrel{def}{=} (free, \mu_trans).E_{i_003}$	Med_{00_01}	$\stackrel{def}{=} (free, \top).Med_{00_01}$
	$+ (occ, \mu_data).E_{i_000};$		$+ (go_next, \mu_slot).Med_{00_02}$
E_{i_003}	$\stackrel{def}{=} (db_i, \top).E_{i_004};$		$+ (send, \top).Med_{00_05};$
E_{i_004}	$\stackrel{def}{=} (free, \mu_trans).E_{i_005}$	Med_{00_02}	$\stackrel{def}{=} (ack_n, \mu_trans).Med_{00_03}$
	$+ (occ, \mu_data).E_{i_000};$		$+ (ack_m, \mu_trans).Med_{00_04};$
E_{i_005}	$\stackrel{def}{=} (send, \mu_trans).E_{i_006};$	Med_{00_03}	$\stackrel{def}{=} (sync_n, \mu_data).Med_{00_00};$
E_{i_006}	$\stackrel{def}{=} (ack_i, \top).E_{i_007}$	Med_{00_04}	$\stackrel{def}{=} (sync_m, \mu_data).Med_{00_00};$
	$+ (collision, \top).E_{i_008};$	Med_{00_05}	$\stackrel{def}{=} (collision, \mu_trans).Med_{00_06};$
E_{i_007}	$\stackrel{def}{=} (succ_i, \mu_trans).E_{i_009};$	Med_{00_06}	$\stackrel{def}{=} (collision, \mu_data).Med_{00_00};$
E_{i_008}	$\stackrel{def}{=} (coll_i, \mu_trans).E_{i_000};$		
E_{i_009}	$\stackrel{def}{=} (sync_i, \top).E_{i_000};$		

Tableau 2. *Nœud*Tableau 3. *Medium*

actions ($succ_i, \mu_trans$) et ($coll_i, \mu_trans$) servent à faire évoluer la composante *backoff* après une collision ou une transmission correcte. L'activité ($sync_i, \top$) est utilisée pour l'évaluation de performances mais n'est pas essentielle.

Modèle de la topologie : La topologie du réseau d'infrastructure est modélisée avec la composante *Med* décrivant le partage du médium (voir tableau 3). Les activités de cette composante sont partagées avec les nœuds, puisque tous les nœuds partagent le même médium. Med_{00_00} représente le médium libre. L'action *free* permet aux nœuds de tester si le médium est libre. Un nœud prend possession du médium avec une action *send*, le médium bascule dans l'état Med_{00_01} . Dans cet état le médium est libre pour une durée μ_slot permettant ainsi de modéliser une collision. Si un autre nœud transmet un paquet pendant cette durée, une collision est signalée. Sinon, un acquittement ack_i est envoyé pour le nœud i . Comme signalé auparavant, $sync_i$ est une action permettant de mesurer un critère de performance. En particulier, nous pouvons ainsi mesurer le taux d'occupation du médium par le nœud i .

Le modèle du système représentant un réseau 802.11 d'infrastructure avec 2 nœuds émetteurs est modélisé par le système suivant :

$$Scenario \stackrel{def}{=} ((E_{0_000} \boxtimes_K BO_{0_0}) || (E_{1_000} \boxtimes_L BO_{1_0})) \boxtimes_M Med_{00_00}$$

Les ensembles de synchronisation sont pour $K = \{db_0, succ_0, coll_0\}$, $L = \{db_1, succ_1, coll_1\}$ et $M = \{send, ack_0, ack_1, sync_0, sync_1, collision\}$. On voit ici clairement que nous pouvons étudier le comportement de 2 nœuds utilisant le même protocole de *backoff* (BEB, DIDD, MILD, ...) ou utilisant chacun un protocole différent (BEB et DIDD, DIDD et MILD, etc.).

Extension à n nœuds : Le modèle décrit précédemment permet de modéliser deux nœuds partageant le médium. L'étude de 3 ou 4 nœuds est encore envisageable, mais passer à l'étude de n nœuds conduit à une explosion de l'espace d'états que même les techniques d'agrégation stochastique ne peuvent résoudre. Nous proposons donc

de modéliser un nœud *charge* simulant le comportement de $n - 1$ nœuds (tableau 4). L'idée est que le nœud charge a le comportement agrégé de $n - 1$ nœuds du réseau. Nous modélisons aussi bien les transmissions de données, les temps d'attentes aléatoires dû au *backoff* que les collisions au sein de ces $n - 1$ nœuds agrégés grâce à une modélisation fine du *backoff* du nœud charge. Le dimensionnement des paramètres pour n nœuds se fait de façon récursive à partir de $n - 1$ nœuds.

Parmi les actions d'un nœud standard, certaines ont été ajoutées, retirées, modifiées. Nous avons ajouté deux actions de transmission correspondant à la génération d'une transmission normal ou d'une collision pouvant se produire parmi les $n - 1$ nœuds modélisé dans le nœud charge. Les taux sont calculés à partir des résultats obtenus récursivement pour $n - i_{i \in [0..n-2]}$. Les synchronisations du *backoff* ont elles aussi été retirées car le tirage du *backoff* dans une fenêtre de contention se fait suivant la distribution des probabilités faites à l'étape précédente. Le modèle PEPA est représenté sur le tableau 5.

E_c_000	$\stackrel{def}{=}$	$(free, \mu_trans).E_c_001;$
E_c_001	$\stackrel{def}{=}$	$(difs, \mu_difs).E_c_002;$
E_c_002	$\stackrel{def}{=}$	$(free, \mu_trans).E_c_003$ $+ (occ, \mu_data).E_c_000;$
E_c_003	$\stackrel{def}{=}$	$(db_i, \top).E_c_004;$
E_c_004	$\stackrel{def}{=}$	$(free, \mu_trans).E_c_005$ $+ (occ, \mu_data).E_c_000;$
E_c_005	$\stackrel{def}{=}$	$(send, \mu_trans_ok).E_c_006$ $+ (send, \mu_trans_col).E_c_000;$
E_c_006	$\stackrel{def}{=}$	$(ack_i, \top).E_c_000$ $+ (collision, \top).E_c_000;$

Tableau 4. Nœud charge

BO_charge	$\stackrel{def}{=}$	$(proba, p_0).BO_i_0..$ $+ (proba, p_j).BO_i_j;$
...		...
BO_i_j	$\stackrel{def}{=}$	$(db_i, f_j).BO_charge;$

Tableau 5. backoff charge

4.3. Topologie multi-sauts

Sur les mêmes principes que le réseau avec infrastructure, nous avons construit les modèles suivant.

Les stations cachées Par manque de place, nous ne pouvons présenter le modèle des noeuds cachés. Nous invitons le lecteur à consulter [RAZ 06] pour plus de détails.

Les 3 paires Dans le scénario des 3 paires, la paire se trouvant au centre *lutte* pour l'accès au médium avec les deux paires extérieures tandis que les paires extérieures ne *luttent* qu'avec la paire centrale. Ainsi, la paire centrale n'accède pas souvent au médium. Dans ce scénario, aucune collision ne peut avoir lieu du fait que chaque émetteur est très proche de son récepteur. Le modèle de nœud pour les 3 paires est exactement le même que pour le réseaux avec infrastructure, seul le médium change car il modélise les interactions entre les noeuds, et donc la topologie. Dans le modèle

de médium, les deux paires extérieures peuvent émettre simultanément et occupent le médium du point de vue de la paire centrale. μ_dataXX modélise le pourcentage du paquet déjà envoyé. Le suffixe m est associé aux actions liées aux paires extérieures tandis que les suffixes n sont liés à la paire centrale.

$$\begin{aligned}
Med_0_000 &= (free_m, \top).Med_0_000 + (free_n, \top).Med_0_000 \\
&\quad + (transmit_m, \top).Med_0_002 + (transmit_n, \top).Med_0_001; \\
Med_0_001 &= (Ack_n, \mu_data100).Med_0_000; \\
Med_0_002 &= (f, \mu_data25).Med_0_003 + (free_m, \top).Med_0_002; \\
Med_0_003 &= (f, \mu_data25).Med_0_004 + (free_m, \top).Med_0_003 \\
&\quad + (transmit_m, \top).Med_0_007; \\
Med_0_004 &= (f, \mu_data25).Med_0_005 + (free_m, \top).Med_0_004 \\
&\quad + (transmit_m, \top).Med_0_008; \\
Med_0_005 &= (Ack_m, \mu_data25).Med_0_000 + (free_m, \top).Med_0_005 \\
&\quad + (transmit_m, \top).Med_0_009; \\
Med_0_007 &= (Ack_m, \mu_data75).Med_0_004; \\
Med_0_008 &= (Ack_m, \mu_data50).Med_0_003; \\
Med_0_009 &= (Ack_m, \mu_data25).Med_0_002;
\end{aligned}$$

5. Évaluation de performances

Les métriques de performances et d'équité : Pour évaluer les performances des stratégies de *backoff* nous utilisons deux métriques. *i)* L'efficacité de l'algorithme est mesurée comme étant le temps d'occupation du médium. Cette mesure est liée au débit. *ii)* L'équité est mesurée grâce à un vecteur de probabilités noté $alpha = (\alpha_i)_{\forall i}$. α_i est la probabilité que la transmission du i^{ieme} paquet soit correcte sachant que les transmissions des $[1..i - 1]^{iemes}$ paquets ont toutes été correctes et proviennent du même émetteur. Cette métrique est une métrique d'équité représentant la probabilité de monopolisation du canal radio par une station.

Validation du modèle : La figure 5 valide notre modèle. Les courbes de simulations et théorique de notre modèle sont proches sauf quand le nombre de stations augmentent. L'écart est dû à la modélisation des collisions qui n'impliquent que deux stations au plus. Sur cette figure, toutes les stations émettent des paquets de 1000 octets à 11Mbps. Ces mêmes hypothèses seront utilisées pour tous les calculs si d'autres hypothèses ne sont pas spécifiées. La figure 6 montrent l'évolution du temps d'occupation correcte, du temps d'occupation par les collisions, et du temps de canal libre en fonction de la taille des paquets et du débit utilisé par les stations. Au final, le critère pertinent n'est pas tant le débit que la longueur des paquets mais bien la durée d'occupation du canal pour transmettre.

En mode ad hoc : Les figures 7 et 8 montrent respectivement les temps d'occupation du canal radio pour le scénario des 3 paires³ et pour les stations cachées. Nous voyons sur la figure 7 que le temps d'occupation maximum pour la paire centrale ne dépasse pas 12% lorsque les 3 stations émettent des paquets de 100 octets. Les courbes de MILD, BEB et DIDD sont identiques, ceci est dû au fait que sur ce scénario il n'y a pas de collision, la fenêtre de contention reste donc toujours la même. La figure 8

3. Tous les résultats des 3 paires concernent la paire centrale

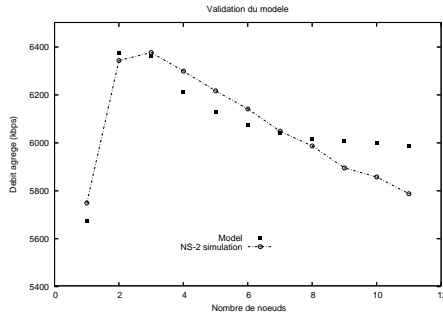


Figure 5. Validation du modèle.

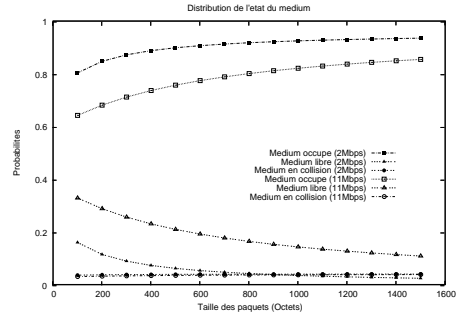


Figure 6. Différent débit.

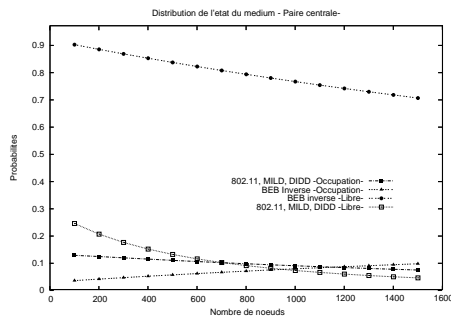


Figure 7. Efficacité sur les 3 paires.

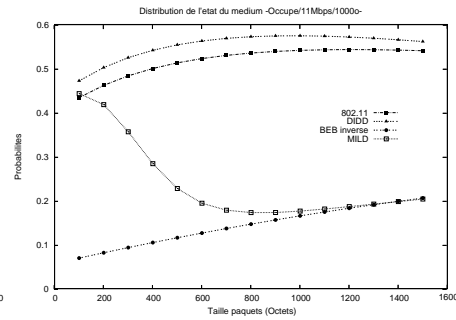


Figure 8. Efficacité : stations cachées

montre l'efficacité sur le scénario des stations cachées. L'algorithme DIDD est le plus efficace, suivi de BEB puis MILD et de BEB Inversé. Avec BEB inversé, plus la taille de paquets augmente, plus l'occupation du canal augmente, alors que pour BEB et DIDD il existe une taille optimale de paquet maximisant le temps d'occupation du canal. En effet, plus les paquets sont grands, plus la probabilité de collision augmente. Pour MILD, le temps d'occupation diminue vite avec la taille de paquet puis augmente linéairement comme BEB Inversé. Ce comportement s'explique par le fait que l'algorithme de *backoff* MILD rejoint rapidement les grands états de *backoff*, d'où le comportement proche de BEB inversé quand le nombre de collisions augmente.

Les figures 9 et 10 montrent les courbes de α pour les scénarios des stations cachées et des 3 paires. La figure 9 montre que les 3 algorithmes de *backoff* BEB, DIDD, et MILD ne sont pas équitables car la probabilité α augmente avec le nombre de transmissions successives. L'algorithme BEB Inversé est le plus équitable avec un α quasiment constant grâce à une grande fenêtre de *backoff* permettant presque l'envoi de paquets de manière alterné. L'algorithme DIDD est le moins équitable (sur les stations cachées) avec un α croissant et élevé. Avec DIDD les stations vont avoir tendance à monopoliser le canal radio (comme BEB à moindre échelle). BEB Inversé est le plus équitable des algorithmes sur ce scénario. La valeur de α pour MILD croît plus lente-

ment que pour BEB et DIDD grâce à la diminution linéaire de la fenêtre de contention.

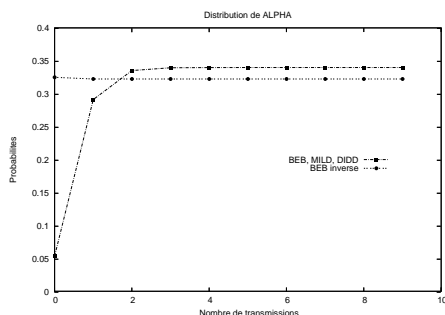


Figure 9. α sur les 3 paires.

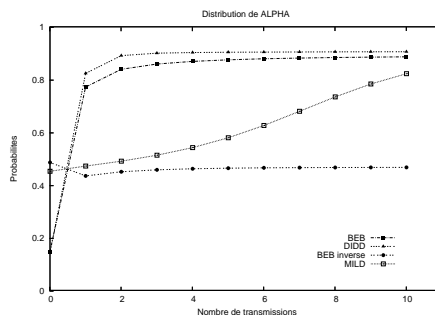


Figure 10. α sur les stations cachées

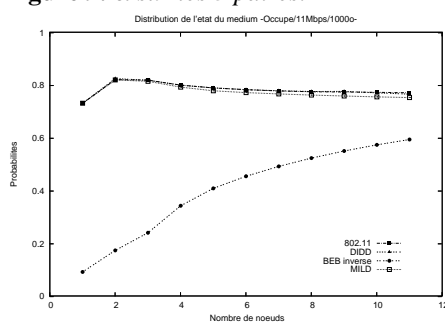


Figure 11. Efficacité en infrastructure.

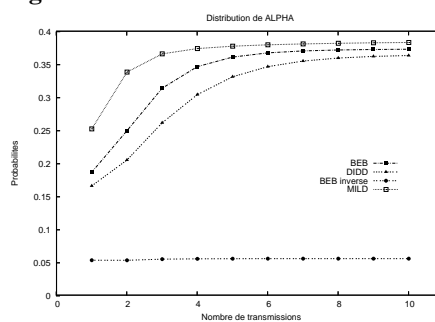


Figure 12. α en infrastructure

En mode infrastructure : La figure 11 montre l'occupation du canal radio en fonction du nombre de stations en mode infrastructure. Les courbes de BEB, DIDD et MILD sont assez proches. Cette similarité est due au fait que l'émission de paquet correcte est très fréquente, ce qui provoque un retour vers les petits états de *backoff*. Pour BEB Inversé l'occupation augmente avec le nombre de stations sur le réseau car plus il y a de stations, plus le médium sera exploité. On pressent que l'algorithme BEB Inversé va atteindre un maximum avant de décroître, ceci étant dû à un trop grand nombre de stations provoquant beaucoup de collisions.

La figure 12 montre l'évolution de α pour les 4 algorithmes de *backoff*, avec 11 stations à 11Mbps et des paquets de 1000 octets en mode infrastructure. On voit que BEB Inversé est l'algorithme le plus équitable avec un α constant. L'algorithme le moins équitable est MILD avec des probabilités α élevées suivi de BEB et DIDD. MILD est l'algorithme le moins efficace dû à la diminution linéaire de la taille de la fenêtre de contention après une transmission acquitée alors qu'une collision augmente rapidement la taille de la fenêtre de contention. Ceci augmente donc fortement la probabilité d'accès des stations ayant déjà accédé au médium.

6. Conclusions et perspectives

Dans cet article, nous proposons l'étude de plusieurs algorithmes de *backoff* sur plusieurs topologies de réseaux sans fil différentes. Le premier résultat qui apparaît est le compromis équité-efficacité. Les topologies et les *backoff* étudiés montrent que plus le *backoff* est efficace moins il est équitable. Nous pouvons, d'après les résultats, aussi montrer que parmi les algorithmes de *backoff* étudiés, aucun n'est efficace pour les 3 paires, surtout au niveau de la paire centrale. On montre aussi que contrairement au cas d'un réseau à infrastructure, l'augmentation de la taille du paquet, dans un réseau multi-sauts, n'augmente pas l'efficacité. On peut déduire de ces résultats, pour un réseau avec infrastructure, que les algorithmes tels que DIDD et BEB ne sont efficaces que pour un petit nombre de stations alors que les algorithmes BEB Inversé et MILD seront efficaces avec un plus grand nombre de stations. MILD est un bon exemple de l'antagonisme des comportements au passage d'un réseau avec infrastructure vers un réseau ad hoc car l'algorithme est efficace à un saut mais ne l'est pas en multi-sauts. De même DIDD, est moins équitable que BEB et MILD dans un contexte multi-sauts, alors que dans un contexte avec infrastructure, DIDD est plus équitable.

Nous avons donc mis en avant le compromis équité-efficacité des algorithmes de *backoff* mais aussi que les contextes multi-sauts et infrastructure requièrent des algorithmes de *backoff* différents. La prochaine étape de ce travail est de caractériser les propriétés de plusieurs algorithmes de *backoff* afin de fournir une méthode, selon les topologies, de construire un algorithme efficace et/ou équitable.

7. Bibliographie

- [BHA 94] BHARGHAVAN V., DEMERS A., SHENKER S., ZHANG L., « MACAW : A Media Access Protocol for Wireless LANs », *ACM SIGCOMM*, London, U.K., Sep 1994.
- [BIA 00] BIANCHI G., « Performance Analysis of the IEEE 802.11 DCF », *IEEE Journal on selected areas in communications*, vol. 18, n° 3, 2000, p. 353-547.
- [CHA 04] CHAUDET C., GUÉRIN LASSOUS I., THIERRY E., GAUJAL B., « Study of the Impact of Asymmetry and Carrier Sense Mechanism in IEEE 802.11 Multi-hops Networks through a Basic Case », *ACM PE-WASUN*, Venice, Italy, Sept. 2004.
- [CHA 05] CHATZIMISIOS P., BOUCOUVALAS A., VITSAS V., VAFIADIS A., OIKONOMIDIS A., HUANG P., « A simple and effective backoff scheme for the IEEE 802.11 MAC protocol », *CITSA*, Orlando, Florida, USA, July 2005.
- [HIL 94] HILLSTON J., « A compositional approach to performance modelling », PhD. Thesis, The University of Edinburgh, 1994.
- [LI 04] LI Z., NANDI S., GUPTA A. K., « Modeling the Short-Term Unfairness of IEEE 802.11 in Presence of Hidden Terminals », *IFIP NETWORKING*, 2004, p. 613-625.
- [M.K 01] M. KWIATKOWSKA G. NORMAN D. P., « PRISM : Probabilistic Symbolic Model Checker », *PAPM/PROBMIV'01 Tools Session*, Durham, North Carolina, USA, 2001.
- [RAZ 06] RAZAFINDRALAMBO T., VALOIS F., « Stochastic behavior study of backoff algorithms in case of hidden terminals », *IEEE PIMRC*, Helsinki, Finland, Sept 2006.
- [S.G 94] S. GILLMORE J. H., « The PEPA workbench : a tool to support a process algebra-based approach to performance modelling », *TOOLS*, Vienna, Austria, May 1994.