



**HAL**  
open science

# Algorithmes distribués d'ordonnement dans les réseaux sans-fil.

Jean-Claude Bermond, Dorian Mazauric, Philippe Nain

► **To cite this version:**

Jean-Claude Bermond, Dorian Mazauric, Philippe Nain. Algorithmes distribués d'ordonnement dans les réseaux sans-fil.. JDIR 2009, Feb 2009, Belfort, France. inria-00505521

**HAL Id: inria-00505521**

**<https://inria.hal.science/inria-00505521>**

Submitted on 3 Sep 2022

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

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

# Algorithmes distribués d'ordonnement dans les réseaux sans-fil

Jean-Claude Bermond<sup>1</sup>, Dorian Mazauric<sup>1,2</sup>, Philippe Nain<sup>2</sup>

<sup>1</sup> MASCOTTE, INRIA, I3S, CNRS, Univ. Nice-Sophia Antipolis

<sup>2</sup> MAESTRO, INRIA

Sophia Antipolis, France

{firstname.lastname@sophia.inria.fr}

**Résumé**—Nous considérons dans cet article le problème d'ordonnement distribué dans les réseaux sans-fil. En raison des interférences dans ce type de réseau, ne peuvent être activés simultanément que des liens n'interférant pas entre eux. Par exemple dans un modèle primaire, on ne peut activer que des liens deux à deux non adjacents. Nous nous plaçons dans un contexte d'arrivée aléatoire de messages et l'objectif est d'assurer un bon comportement du réseau en particulier d'assurer la stabilité des files d'attente, en limitant le nombre moyen de messages en attente. Des algorithmes centralisés permettant de décider quels liens sont activés à chaque étape existent mais ils supposent une connaissance globale du réseau et sont peu adaptés aux applications. Il est donc nécessaire de concevoir des algorithmes distribués qui utilisent une connaissance très locale du réseau. Nous proposons dans cet article deux algorithmes distribués, valides quel que soit le modèle d'interférence binaire et avec une phase de contrôle de durée constante, améliorant les algorithmes existants vérifiant uniquement l'un ou l'autre de ces deux critères.

**Index Terms**—réseau, sans-fil, ordonnancement, distribué, algorithme, interférence, stabilité.

## I. INTRODUCTION

Le problème d'ordonnement des appels est un problème fondamental dans les réseaux de télécommunication. Il a été très largement étudié dans les réseaux filaires avec notamment la condition évidente que la demande de trafic sur un lien ne doit pas excéder sa capacité. Pour les réseaux sans-fil (de type radio, ad hoc ou senseurs par exemple), une contrainte supplémentaire vient des problèmes d'interférence. En effet lors d'une étape d'envoi de données, il n'est pas possible d'activer certains liens de communication du réseau sans-fil simultanément, car dans ce cas des interférences empêcheraient la bonne réception de ces différents envois. Ainsi il est indispensable avant toute phase d'envoi de choisir un ensemble de liens vérifiant les contraintes du modèle d'interférence. En respectant le fait que ces liens choisis soient les seuls à envoyer des données au cours de cette étape, il est alors garanti qu'aucune interférence nuira à ces envois de données. Etant donné une charge de trafic sur chaque lien, suivant une certaine loi aléatoire dans notre modèle, l'objectif est d'écouler les messages en attente d'envoi en ordonnant les liens au cours des différentes étapes et en respectant les contraintes d'interférence. Pour résoudre ce problème d'ordonnement, des algorithmes centralisés optimaux sont proposés dans [1] [2] pour des arrivées de messages stochastiques et dans [3] pour

des arrivées de messages prédéterminées. Mais en pratique, les algorithmes utilisés doivent être distribués. Pour des réseaux de capteurs par exemple, la décision (d'émettre ou pas) doit être prise localement, sans passer par une station centrale, dans le but d'économiser de l'énergie notamment. Une des plus importantes difficultés est donc de prendre en compte les interférences spécifiques aux réseaux sans-fil dans le but de concevoir des algorithmes locaux valides quel que soit le modèle d'interférence et avec une phase de contrôle de durée constante. Les travaux présentés notamment dans [4] [5] [6] décrivent des algorithmes distribués avec un nombre de « time slots » de contrôle croissant avec la taille du réseau.

Dans ce contexte il s'agit que durant une phase d'envoi de données, les appels n'interfèrent pas entre eux. Nous verrons (Section II) comment modéliser formellement les interférences entre les liens de communication et nous décrirons les algorithmes que nous proposons pour ce problème (Section IV). Nous considérons des réseaux synchrones et nous divisons le temps en étapes, chacune de même taille. Durant une étape, une première phase de contrôle permet de déterminer quelles vont être les liens activés durant la seconde phase, la phase d'envoi de données. L'objectif est d'activer les liens ayant un nombre important de messages en attente. Dans le cas général (modèle d'interférence différent du modèle primaire), le problème de trouver un ensemble d'arêtes respectant les contraintes d'interférence et maximisant la somme des poids est un problème *NP-Complet*. Les algorithmes proposés devront être distribués (locaux), la phase de contrôle sera de durée constante et les algorithmes valides quel que soit le modèle d'interférence binaire. Nous supposons que le trafic est « single-hop » : un message envoyé quitte le réseau instantanément après même si la généralisation à un trafic multi-hop est possible. Les algorithmes doivent alors garantir que le nombre moyen de messages en attente sur chaque lien du réseau soit borné.

La Figure 1(a) représente un réseau sans-fil de quatre stations, modélisé par un graphe, où les arêtes représentent les différents appels possibles, les différents liens de communication. Le modèle d'interférence « primary node » contraint à avoir un ensemble de liens activés formant un couplage dans le graphe associé (un couplage est un ensemble d'arêtes deux à deux disjointes). Nous supposons dans cet exemple que durant une étape, les liens activés peuvent écouler un unique

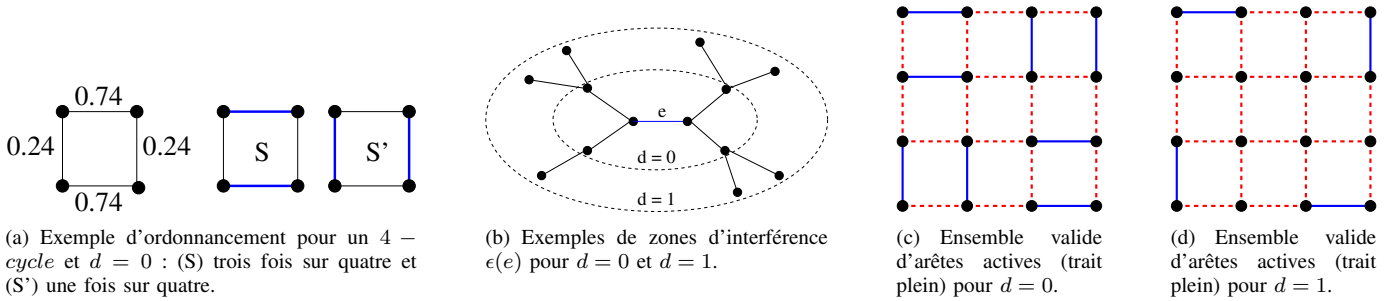


FIG. 1. Exemples d'ordonnement simple (a), de zones d'interférence (b) et d'ensembles valides d'arêtes actives (c,d).

paquet et les valeurs sur les liens représentent le nombre moyen d'arrivées de paquets au cours d'une étape. Ainsi un ordonnancement trivial est d'activer trois fois sur quatre les deux arêtes de  $S$  et une fois sur quatre les deux arêtes de  $S'$ . La taille moyenne des files d'attente sur les liens est bornée.

L'article est organisé comme suit. Nous présentons tout d'abord le modèle utilisé dans nos études (Section II) avant de présenter brièvement quel ques travaux existants (Section III). Dans la Section IV, nous proposons deux algorithmes distribués d'ordonnement dans les réseaux sans-fil, vérifiant les critères décrits précédemment. Enfin nous analysons ces algorithmes dans la Section V avant de présenter quel ques résultats de simulation dans la Section VI.

## II. MODÈLE

Un réseau sans-fil est modélisé par un graphe (non orienté) des transmissions  $G = (V, E)$  où les sommets et les arêtes représentent respectivement les stations et les liens de communication. Autrement dit, deux sommets  $u, v \in V$  sont reliés si et seulement si  $u$  (respectivement  $v$ ) peut transmettre un paquet à  $v$  (respectivement  $u$ ). Nous supposons dans ce modèle que les liens de communication sont symétriques, comme dans les réseaux radio par exemple avec l'utilisation d'accusés de réception. Les interférences peuvent se modéliser de différentes manières en utilisant par exemple le rapport signal sur bruit. Mais cette représentation est difficile à analyser et ainsi nous les modélisons en pratique par des modèles binaires. Définissons formellement la zone d'interférence d'une arête comme suit :

*Définition 1:* Etant donné un graphe des transmissions  $G = (V, E)$ , une arête  $e \in E$  et un entier positif ou nul  $d$ , la zone d'interférence de  $e = u_1 u_2$  est  $\epsilon(e) = \{e' = v_1 v_2 \in E, \exists i, j \in [1, 2], D(u_i, v_j) \leq d\}$ .  $D$  est la distance dans  $G$ , c'est-à-dire la longueur d'un plus court chemin dans  $G$ .

Les modèles d'interférence seront définis par une valeur de  $d$  fixée. Le cas particulier  $d = 0$  représente le modèle d'interférence « primary node » qui consiste à déterminer un couplage dans le graphe des transmissions comme dans la Figure 1(a). Cependant les modèles les plus réalistes sont définis pour  $d > 0$ . La Figure 1(b) montre un exemple de zones d'interférence pour  $d = 0$  et  $d = 1$ . Ainsi étant donné un modèle d'interférence ( $d$  fixé), deux arêtes actives à une étape

$t$ , ne doivent pas se situer dans une même zone d'interférence.

En considérant des réseaux synchrones, le temps est divisé en différentes étapes de même durée. Une étape contient alors deux phases distinctes : une phase de contrôle où il s'agit de déterminer quelles seront les liens de communication (arêtes du graphe des transmissions) autorisés à envoyer des données durant la seconde phase (phase d'envoi).

Nous introduisons pour chaque arête  $e \in E$  et pour chacune des étapes  $t \geq 0$ , la variable  $a_t(e)$  telle que  $a_t(e) = 1$  si  $e$  est active, autorisée à envoyer des messages lors de l'étape  $t$  ( $a_t(e) = 0$  sinon). En intégrant la modélisation des interférences décrite précédemment, durant une étape deux arêtes autorisées à envoyer des données ne doivent pas être dans une même zone d'interférence. Formellement  $\forall e, e' \in E$  si  $a_t(e) = a_t(e') = 1$  alors  $e' \notin \epsilon(e)$  et donc  $e \notin \epsilon(e')$ . La Figure 1(c) (respectivement Figure 1(d)) décrit un exemple d'ensemble valide maximal d'arêtes actives (aucune autre arête ne peut devenir active sans changer cet ensemble) pour le modèle d'interférence  $d = 0$  (respectivement  $d = 1$ ). Cela correspond à un couplage (respectivement un couplage induit) dans le graphe des transmissions.

Le trafic étant supposé « single-hop », un paquet envoyé via un lien de communication, quitte instantanément le réseau une fois sa réception terminée. Il est donc possible d'associer un paquet à un lien du réseau sans-fil, correspondant à une arête du graphe des transmissions. Ainsi à chaque lien du réseau est associée une file d'attente contenant tous les paquets qui doivent être transmis via ce lien. La politique de service peut être quelconque, les concepteurs du réseau la choisiront selon le type d'application par exemple. Nous notons par  $q_t(e)$  le nombre de messages en attente dans la file d'attente du lien  $e \in E$ . Au cours de chaque étape, de nouveaux paquets arrivent sur les liens du réseau (les arêtes du graphe) selon une certaine loi de probabilité propre en général à chaque lien et pas nécessairement connue. Nous notons  $A_t(e)$  le nombre de messages arrivant sur  $e \in E$  au cours de l'étape  $t \geq 0$ . Remarquons que ces nouveaux paquets sont réellement pris en compte à l'étape  $t + 1$ . Enfin chaque lien de communication a une capacité propre qui est en fait le nombre de paquets qu'il peut envoyer durant une phase d'envoi de données (si le lien est actif évidemment). Nous notons cette capacité  $c(e)$ . Ainsi le poids d'une arête vérifie, en notant  $[x]^+ = \max(x, 0)$ ,  $q_{t+1}(e) = (q_t(e) - a_t(e) \cdot c(e))^+ + A_t(e)$ .

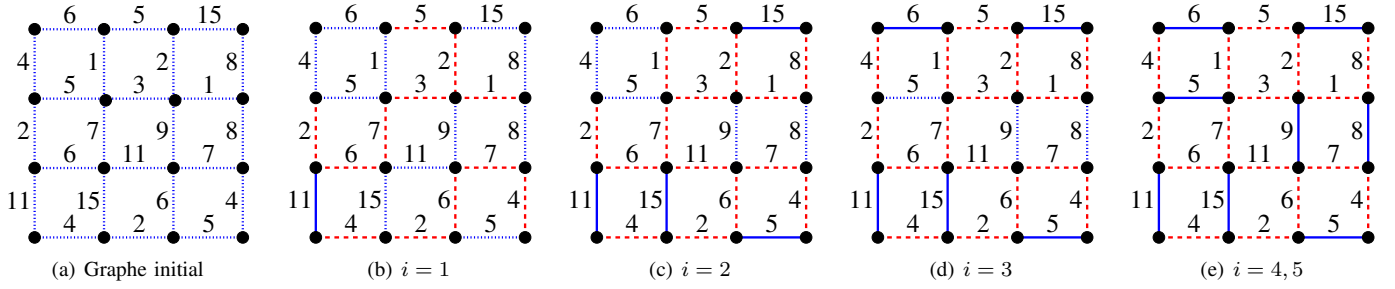


FIG. 2. Sous-phase régulière de Algorithme Log 1 pour une grille  $4 \times 4$ , pour  $d = 0$  et  $K = 16$  (5 time slots de contrôle). Les arêtes de trait plein sont actives, les tirets représentent les inactives et les pointillés sont les arêtes encore indéterminées.

### III. TRAVAUX EXISTANTS

Nous présentons brièvement dans cette partie deux algorithmes proposés dans [7] et [8]. Ils sont tous les deux distribués mais l'algorithme proposé dans [7], n'admet pas une phase de contrôle de durée constante alors que celui décrit dans [8] est valable uniquement si le modèle d'interférence est le « primary node » ( $d = 0$ ).

#### A. Algorithme Q-SCHED

Dans [7], les auteurs présentent un algorithme distribué pour le problème d'ordonnancement dans les réseaux sans-fil. Il est valable quel que soit le modèle d'interférence ( $\forall d \geq 0$ ) mais la durée de la phase de contrôle n'est pas constante. Le réseau étant supposé synchrone, chacune des arêtes choisit une valeur de backoff  $t_0$  au début de chaque phase de contrôle, c'est-à-dire la date d'émission  $t_0$  de son unique message de contrôle. Si elle reçoit un message avant ou durant  $t_0$ , l'arête sera inactive durant la prochaine phase d'envoi de données. Sinon, elle sera active. Cet algorithme simple est valable pour  $d$  quelconque. En revanche le choix du backoff d'une arête  $e \in E$  est fonction du poids de chacune des arêtes de sa zone d'interférence  $\epsilon(e)$  et du poids de chacune des arêtes appartenant à  $\epsilon(e')$  avec  $e' \in \epsilon(e)$ . Une arête doit ainsi mettre à jour, à chaque étape, le poids de toutes les arêtes situées dans son  $2d - \text{voisinage}$ . Ainsi les problèmes d'interférence se posent pour les mises à jour. Il est donc impossible de déployer cet algorithme en pratique.

#### B. Algorithme de Chemins Augmentant

L'algorithme décrit dans [8] a lui un nombre constant de time slots de contrôle, mais n'est valide que si le modèle d'interférence est le « primary node » ( $d = 0$ ). L'idée est de trouver lors d'une étape, un nouveau couplage améliorant le précédent en terme de poids total des arêtes choisies. Etant donné un couplage  $M_t$  à l'étape  $t$ , durant l'étape  $t + 1$  l'algorithme va construire des chemins alternant arêtes appartenant à  $M_t$  et arêtes n'y appartenant pas. A la fin, si le poids total des arêtes inactives sur un chemin alternant est plus grand que le poids des arêtes de ce chemin appartenant à  $M_t$ , alors le couplage est modifié, le rôle de chacune des arêtes du chemin est inversé. Cette phase de contrôle est de durée constante mais cette méthode n'est possible que si le modèle d'interférence

est le modèle « primary node », le moins réaliste. De plus l'algorithme converge très lentement.

Il est donc nécessaire de concevoir des algorithmes distribués (ou locaux) avec une phase de contrôle de durée constante, valides quel que soit le modèle d'interférence ( $\forall d \geq 0$ ) et convergant rapidement.

### IV. ALGORITHMES PROPOSÉS

Nous proposons dans cette partie deux algorithmes distribués d'ordonnancement dans les réseaux sans-fil, valides quel que soit le modèle d'interférence binaire et assurant une phase de contrôle de durée constante. Comme expliqué précédemment, un réseau est modélisé par un graphe des transmissions valué  $G = (V, E)$ . Nous décrivons les algorithmes dans le cas où les valeurs de  $q_t(e)$  sont dans l'ensemble fini  $[0, K - 1]$  où  $K$  dépend du nombre de time slots de contrôle disponibles. Le cas général (valeurs quelconques de  $q_t(e)$ ) se ramène aisément à ce dernier en partitionnant les valeurs de  $q_t(e)$  en  $K$  classes. Par exemple, nous pouvons mettre les  $q_t(e) \geq M$  ( $M$  une constante) dans la  $K$ -ième classe et en partitionnant les autres valeurs en  $K - 1$  classes de taille à peu près égale. Dans la suite nous supposons que  $\forall t \geq 0, \forall e \in E, q_t(e) \in [0, K - 1]$ .

#### A. Algorithme Log 1

Comme nous l'avons expliqué précédemment, nous considérons des réseaux synchrones et dans ce contexte, le temps est divisé en étapes, chacune de même taille et divisée en deux phases : une phase de contrôle et une phase d'envoi des données. Nous allons décrire précisément l'Algorithme Log 1 qui permet de décider durant cette phase de contrôle quelles seront les liens activés durant la prochaine phase d'envoi de données. Nous divisons la phase de contrôle en deux sous-phases : une *sous-phase régulière* composée de  $T$  time slots et une *sous-phase aléatoire*. Nous décrivons tout d'abord la sous-phase régulière avant de motiver l'ajout de la sous-phase aléatoire. Définissons trois statuts possibles pour une arête au cours de cette première sous-phase de contrôle :

*Définition 2:* Etant donné un graphe des transmissions  $G = (V, E)$ , une arête  $e \in E$  peut être :

- active,  $e$  enverra des données durant la phase d'envoi ;
- inactive,  $e$  n'en enverra pas ;
- indéterminée si cette décision n'est pas encore prise.

TAB. I  
VECTEUR DE CONTRÔLE  $v = v_{t,e}$  DE L'ARÊTE  $e \in E$  À L'ÉTAPE  $t \geq 0$   
POUR CHAQUE POIDS POSSIBLE  $q_1 = q_t(e)$  POUR ALGORITHME LOG 1  
( $K = 16$ ) ET  $q_2 = q_t(e)$  POUR ALGORITHME LOG 2 ( $c_2 = 2$ ,  
 $K' = 2K = 16$  ET  $g = g(t, e)$ ).

$q_1$	$q_2$	$g$	$q'_2$	$v(1)$	$v(2)$	$v(3)$	$v(4)$	$v(5)$
0	0	0	0	0	0	0	0	1
1	0	1	1	0	0	0	1	0
2	1	0	2	0	0	1	0	1
3	1	1	3	0	0	1	1	0
4	2	0	4	0	1	0	0	1
5	2	1	5	0	1	0	1	0
6	3	0	6	0	1	1	0	1
7	3	1	7	0	1	1	1	0
8	4	0	8	1	0	0	0	1
9	4	1	9	1	0	0	1	0
10	5	0	10	1	0	1	0	1
11	5	1	11	1	0	1	1	0
12	6	0	12	1	1	0	0	1
13	6	1	13	1	1	0	1	0
14	7	0	14	1	1	1	0	1
15	7	1	15	1	1	1	1	0

Au début de chaque sous-phase régulière de contrôle, toutes les arêtes sont indéterminées et à la fin, une arête est soit active, soit inactive. Décrivons à présent le protocole utilisé par une arête encore indéterminée  $e \in E$  durant un time slot  $i$ ,  $1 \leq i \leq T$  :

- a) Si  $e$  n'envoie pas de message de contrôle et en reçoit au moins un d'une arête de sa zone d'interférence, alors  $e$  devient inactive ;
- b) Si  $e$  envoie un message de contrôle et n'en reçoit aucun en provenance des arêtes de sa zone d'interférence  $\epsilon(e)$ , alors  $e$  devient active ;
- c) Sinon,  $e$  reste indéterminée.

Après le time slot de contrôle  $T$ , chaque arête indéterminée devient inactive. Expliquons à présent la politique d'envoi de message de contrôle. Chaque arête  $e \in E$  calcule un *vecteur de contrôle*  $v_{t,e} = (v_{t,e}(1), v_{t,e}(2), \dots, v_{t,e}(T))$  où  $v_{t,e}(i) = 1$  ou  $0$ . Si  $v_{t,e}(i) = 1$ , l'arête  $e$  enverra un message lors du time slot de contrôle  $i$  si elle est encore indéterminée, sinon elle n'en enverra pas. Les  $v_{t,e}(i)$  ( $i = 1..T-1$ ) correspondent au  $i$ -ème bit de son poids  $q_t(e)$  écrit en base 2, en ajoutant d'éventuels zéros devant. Remarquons que les arêtes indéterminées de poids impair envoient un message de contrôle au time slot  $T-1$ . Pour ne pas défavoriser les arêtes de poids pair, le time slot  $T$  est réservé pour les arêtes de poids pair encore indéterminées :  $v_{t,e}(T) = 1$  si  $q_t(e)$  est pair,  $v_{t,e}(T) = 0$  sinon. Comme nous avons supposé que le poids des arêtes était borné par  $K$ , nous pouvons écrire pour chaque poids possible la politique d'envoi de messages de contrôle. Par exemple dans la Table I, le vecteur de contrôle est décrit pour tous les poids possibles  $q_1$  si  $K = 16$  (et donc  $T = 5$ ).

La Figure 2 représente la sous-phase régulière de contrôle de l'Algorithme Log 1 pour une grille  $4 \times 4$ , avec  $d = 0$  et  $K = 16$ . Dans cet exemple le couplage ( $d = 0$ ) est maximal, c'est-à-dire qu'aucune arête inactive ne peut devenir active sans créer une interférence. En revanche, dans l'exemple de la Figure 3(a-e) (où la sous-phase régulière de l'Algorithme

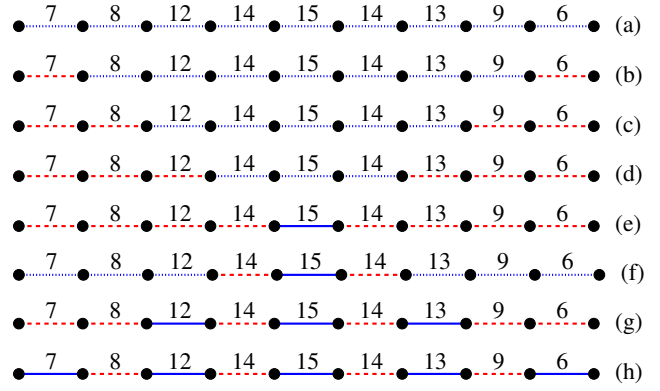


FIG. 3. Algorithme Log 1 (a-e) et Algorithme Log 2 (a-h), pour un chemin avec  $d = 0$  et  $K = 16$ .

Log 1 est appliquée sur le chemin), seule l'arête de poids 15 est active. Dans le but de palier ce problème d'ensemble non maximal, nous ajoutons un nombre supplémentaire constant de time slots de contrôle. Par exemple les arêtes inactives n'ayant pas d'actives dans leurs zones d'interférence redeviennent indéterminées (ceci est possible avec un time slot de contrôle) (Figure 3(f)) et choisissent ensuite aléatoirement un nouveau backoff. Ainsi dans l'exemple de la Figure 3(e), les time slots supplémentaires peuvent permettre par exemple d'activer aussi les liens 8, 13, 6. En pratique 3 time slots aléatoires supplémentaires suffisent.

*Lemme 1:* La durée de la phase contrôle de l'Algorithme Log 1 est  $T_1 = \lceil \log_2(K) \rceil + 4 = O(\log_2(K))$ .

### B. Algorithme Log 2

Algorithme Log 2 est directement adapté de l'Algorithme Log 1 et permet de palier plus efficacement le problème d'ensemble non maximal à la fin de la sous-phase régulière de l'Algorithme Log 1. Au lieu de choisir aléatoirement un nouveau backoff, on répète plusieurs fois cette sous-phase régulière sur l'ensemble des arêtes inactives n'ayant pas d'actives dans leur zone d'interférence. La Figure 3 représente un chemin de 10 sommets, où la sous-phase régulière de l'Algorithme Log 1 active seulement le lien de poids 15 (Figure 3(a-e)) et ainsi le couplage ( $d = 0$  dans cet exemple) n'est pas maximal. En répétant une première fois cette sous-phase régulière, nous obtenons le couplage de la Figure 3(g). En la répétant une seconde fois nous avons un couplage maximal (Figure 3(h)).

Pour prendre en compte les arêtes de même poids, nous avons choisi de définir un nouveau poids (virtuel)  $q'_t(e)$  pour chacun des liens  $e \in E$  tel que  $q'_t(e) = c_2 \cdot q_t(e) + g(t, e)$  avec  $0 \leq g(t, e) < c_2$ , où  $c_2$  est une constante. Plus précisément  $g(t + t_0, e) = g(t, e) + \text{mod}(t_0, c_2)$  avec  $g(0, e)$  une constante ( $0 \leq g(0, e) < c_2$ ). En prenant en compte ces nouveaux poids, nous répétons  $n$  fois la sous-phase régulière de contrôle de l'Algorithme Log 1 (la valeur de  $n$  sera définie par la suite). En supposant que  $q_t(e) < K$ , nous avons  $q'_t(e) < K' = c_2 K$ . La Table I représente les vecteurs de contrôle pour chaque poids virtuel possible  $q'_2$  avec  $K = 8$  et  $c_2 = 2$ . Cela peut

correspondre à un chemin et au modèle d'interférence  $d = 0$  tel que  $\forall e \in E$  et  $\forall e' \in \epsilon(e)$ , alors  $g(0, e) \neq g(0, e')$ . Dans ce cas, il est certain que deux arêtes appartenant à une même zone d'interférence peuvent être comparées (leurs poids virtuels sont différents). Nous verrons dans la Section V que pour toute arête  $e \in E$ , alors il existe une arête  $e' \in \epsilon(e) \cup \{e\}$  telle que  $a_t(e') = 1$ . Enfin il est possible de remarquer que cette transformation ne change pas l'ordre des poids :  $\forall e, e' \in E$ , si  $q_t(e) \leq q_t(e')$ , alors  $q'_t(e) \leq q'_t(e')$ . D'après les Lemmes 4 et 5 de la Section V,  $n = \text{Log}_2(K')$  suffit.

*Lemme 2:* La durée de la phase contrôle de Algorithme Log 2 est  $T_2 = O(\log_2(K)^2)$ .

La Figure 4 montre un exemple de transformation avec  $c_2 = 2$  (Figure 4(b)). La Figure 4(f) montre les 3 arêtes actives après la première sous-phase régulière. La Figure 4(g) (respectivement 4(h)) montre l'ensemble des appels autorisés formant un couplage maximal après la phase de contrôle de Algorithme Log 2 pour  $q'_t(e)$  (respectivement pour  $q_t(e)$ ). Dans la prochaine section, il est intéressant d'analyser ces algorithmes distribués, notamment en prouvant leurs stabilités.

## V. ANALYSE

Dans cette section, nous analysons formellement les algorithmes proposés précédemment, dans le but notamment de mettre en exergue des conditions suffisantes de stabilité.

*Définition 3:* Etant donné un graphe des transmissions valué  $G = (V, E)$ , on dit que  $e \in E$  est maximum locale stricte à une étape  $t \geq 0$  si et seulement si  $\forall e' \in \epsilon(e)$ ,  $q_t(e') < q_t(e)$  ou  $q'_t(e') < q'_t(e)$ .

*Lemme 3:* Une arête maximum locale stricte est toujours active avec Algorithme Log i ( $i=1,2$ ).

*Preuve :* Par construction de nos algorithmes, une arête est inactive si elle a reçu au moins un message pendant un time slot de contrôle en provenance d'une des arêtes de sa zone d'interférence sans en émettre un elle même ou si elle est restée indéterminée après le dernier message de contrôle. Dans ce cas précis, l'arête ne peut pas être maximum locale stricte et il y a donc une contradiction. Ainsi une arête maximum locale stricte est nécessairement active.

*Lemme 4:* Soit  $G(V, E)$  un graphe des transmissions valué tel que si  $e' \in \epsilon(e)$ , alors  $q_t(e) \neq q_t(e')$ . A la fin de la sous-phase régulière de contrôle de Algorithme Log 1 à l'étape  $t$ , le diamètre maximum parmi les sous-graphes induits par les arêtes inactives dans  $G$  est au plus  $\log_2(K)$ .

*Preuve :* Sans perte de généralité, un des pires des cas peut être représenté par un chemin pour lequel il s'agit de maximiser le diamètre d'un sous-graphe induit par les arêtes inactives après cette sous-phase régulière. Les arêtes du chemin  $e_{-n}, e_{-(n-1)}, \dots, e_{-1}, e_0, e_1, \dots, e_{n-1}, e_n \in E$  sont telles que :

$$\begin{aligned} \forall i \in [-(n-1); n-1], \text{ nous avons} \\ v_{e_{-n}}(1) = v_{e_n}(1) = 0 = 1 - v_{e_i}(1) \text{ et;} \\ \forall i \in [-(n-2); n-2], \text{ nous avons} \\ v_{e_{-(n-1)}}(2) = v_{e_{n-1}}(2) = 0 = 1 - v_{e_i}(2) \text{ et;} \end{aligned}$$

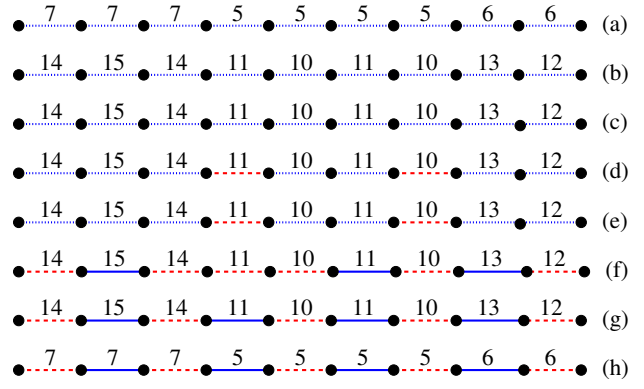


FIG. 4. Algorithme Log 2 pour un chemin de 10 sommets,  $d = 0$ ,  $K = 8$  et  $c_2 = 2$  ( $K' = 16$ ).

...  
 $\forall i \in [-(n-j); n-j]$ , nous avons  
 $v_{e_{-(n-j)}}(j) = v_{e_{n-j}}(j) = 0 = 1 - v_{e_i}(2)$  et ;  
 ...  
 $v_{e_{-2}}(n-1) = v_{e_2}(n-1) = 0 = 1 - v_{e_{-1}}(n-1) =$   
 $1 - v_{e_0}(n-1) = 1 - v_{e_1}(n-1)$  et ;  
 $v_{e_{-1}}(n) = v_{e_1}(n) = 0 = 1 - v_{e_0}(n)$ .  
 Dans ce cas l'unique arête active est  $e_0$  et  $n = \log_2(K)$  maximise le diamètre maximum parmi les sous-graphes induits valant  $\log_2(K)$ .

*Lemme 5:* Etant donné un graphe des transmissions  $G = (V, E)$ ,  $\forall e \in E, \forall t \geq 0, \exists e' \in \epsilon(e) \cup \{e\}$  telle que  $a_t(e') = 1$ , après une phase de contrôle de Algorithme Log 2.

*Preuve :* Tout d'abord nous pouvons transformer les poids de chacune des arêtes  $e \in E$  comme décrit précédemment pour s'assurer que  $\forall e \in E, \forall e' \in \epsilon(e)$ , nous avons  $q'_t(e) \neq q'_t(e')$ . Ensuite à partir du Lemme 4 appliqué à  $q'_t(e)$  et par induction sur le diamètre maximum des sous-graphes induits par les arêtes inactives, il est évident de remarquer qu'après la phase de contrôle de l'étape  $t$  de Algorithme Log 2 ( $n = \log_2(K')$ ),  $\forall e \in E, \exists e' \in \epsilon(e) \cup \{e\}$ , alors  $a_t(e') = 1$ .

*Corollaire 1:* Pour tout  $e \in E$ , nous notons  $a_t(\epsilon(e))$ , la variable aléatoire comptant le nombre d'arêtes actives dans la zone d'interférence de  $e$  à une étape  $t$  donnée. D'après le Lemme 5, nous avons :  $\forall e \in E, E[a_t(\epsilon(e))] \geq 1$  pour Algorithme Log 2.

*Theorème 1:* Soient les hypothèses :  
 $(H_1) \overline{A}(e) := E[A_t(e)] < \infty, \overline{A^2}(e) := E[A_t^2(e)] < \infty.$   
 $(H_2)$  Pour chaque lien du réseau  $e \in E, \overline{A}(e) < c(e)G(e)$  avec  $G(e) = \lim_{K \rightarrow \infty} \inf_{|q| \geq K} \sum_{a(e)=1, a \in \{0,1\}^{|E|}} I(a, q)$  avec  $I(a, q) = P(a_t = a | q_t = q)$ .  
 Sous les hypothèses  $(H_1)$  et  $(H_2)$  et si l'ensemble des arêtes actives  $\{I(\cdot; q), q \in N^{|E|}\}$  est tel que la chaîne de Markov  $\{q_t, t \geq 1\}$  représentant les poids des arêtes soit irréductible, alors la chaîne de Markov est stable.

La preuve complète s'obtient en appliquant le critère de Foster [9] avec comme fonction de Lyapounov  $\sum_{e \in E} q_t(e)^2$  et se trouve dans le rapport en préparation (voir [10]).

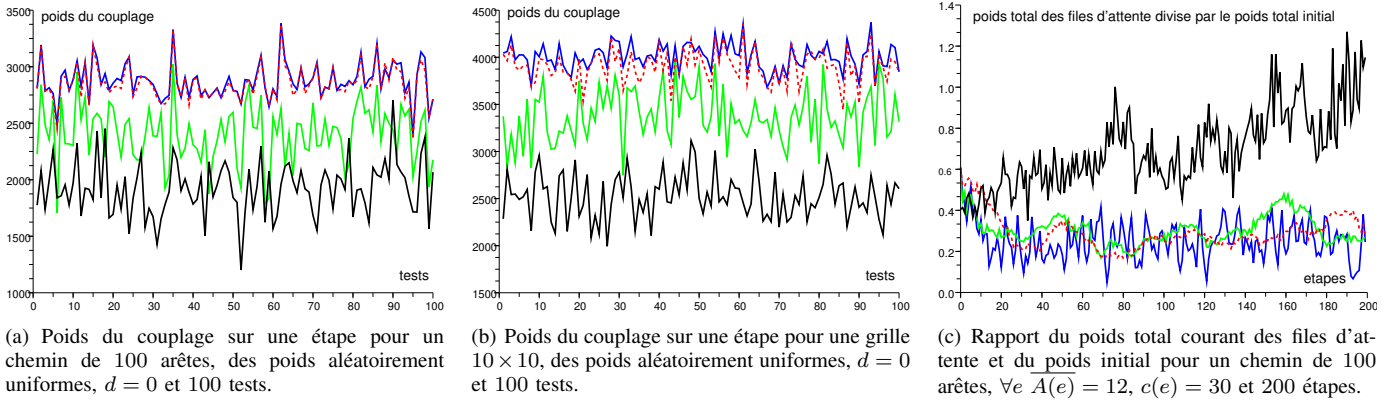


FIG. 5. Résultats de simulations : algorithme centralisé (bleu), Algorithme Log 1 (vert), Algorithme Log 2 (rouge pointillé) et un algorithme aléatoire (noir).

## VI. SIMULATIONS

Nous présentons dans cette partie quelques résultats de simulation. Nous avons tout d'abord implémenté des topologies classiques telles que le chemin et la grille, ainsi que des graphes aléatoires représentant bien les réseaux sans-fil réels [10]. Nous avons implémenté des algorithmes centralisés ainsi que des algorithmes distribués comme l'algorithme de Chemins Augmentant de [8], les deux Algorithmes Log présentés dans la Section IV et un algorithme aléatoire. Les résultats de simulation seront présentés pour le modèle d'interférence binaire  $d = 0$ .

### A. Efficacité sur une étape

Nous décrivons ici les performances des différents algorithmes sur une étape (une phase de contrôle et une phase d'envoi de données) étant donné une certaine distribution des poids des arêtes (tailles des files d'attente). Nous comparons la somme des poids des arêtes du couplage (arêtes actives). Les Figures 5(a) et 5(b) représentent le poids du couplage pour chacun des algorithmes pour 100 tests avec des poids aléatoirement uniformes, respectivement pour un chemin de 100 arêtes et pour une grille  $10 \times 10$ . La sous-phase régulière de l'Algorithme Log 1 donne un poids total proche de celui réalisé par l'algorithme centralisé (Figure 5(a)). Avec 3 time slots aléatoires supplémentaires, l'Algorithme Log 1 donne également un poids proche pour la grille (Figure 5(b)). L'Algorithme Log 2 donne quasiment le même poids que le poids du centralisé pour les deux topologies.

### B. Résultats de Stabilité

Enfin nous étudions la stabilité des files d'attente du réseau en simulant plusieurs étapes consécutives avec des arrivées de paquets au cours de chacune d'elles. La Figure 5(c) montre la stabilité des Algorithmes Log et de l'algorithme centralisé pour un chemin de 100 arêtes avec une capacité  $c(e) = 30$  pour chaque lien  $e \in E$ , un nombre moyen d'arrivées de 12 paquets (uniformément entre 0 et 24) et 200 étapes au total. L'algorithme aléatoire semble lui ne pas être stable. Enfin les résultats de simulation donnent aussi une estimation de la valeur de  $G(e)$  du Théorème 1.

## VII. CONCLUSION

Nous avons proposé dans cet article, deux algorithmes distribués d'ordonnancement dans les réseaux sans-fil, valables quel que soit le modèle d'interférence binaire et avec une phase de contrôle de durée constante. Il serait maintenant intéressant de déterminer analytiquement la valeur de  $G(e)$  du Théorème 1. Enfin, nous pouvons adapter ces algorithmes avec des files d'attente situées sur les sommets du graphe des transmissions et en prenant également en compte un trafic de type « multi-hop ».

### REMERCIEMENTS

Ce travail est en partie supporté par la région PACA et le projet européen IST FET AEOLUS.

### RÉFÉRENCES

- [1] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 1067–1073, May 1997.
- [2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *Proceedings of the 29th IEEE Conference on Decision and Control*, pp. 2130–2132 vol.4, Dec 1990.
- [3] R. Klasing, N. Morales, and S. Pérennes, "On the complexity of bandwidth allocation in radio networks," *Theoretical Computer Science*, vol. 406, no. 3, pp. 225 – 239, 2008, algorithmic Aspects of Global Computing.
- [4] A. Brzezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks : a partitioning approach," in *MobiCom '06*. New York, NY, USA : ACM, 2006, pp. 26–37.
- [5] A. Eryilmaz, O. Asuman, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," *INFOCOM 2007*, pp. 499–507, May 2007.
- [6] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 27–38, 2006.
- [7] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," in *INFOCOM, 2007*, pp. 1631–1639.
- [8] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Sigmetrics, 2007*, pp. 313–324.
- [9] S. Asmussen, *Applied Probability and Queues 2nd Ed.*, Springer, Ed., 2003.
- [10] J.-C. Bermond, D. Mazaauric, and P. Nain, "Distributed call scheduling in wireless networks," INRIA, Tech. Rep. RR-6763, Dec. 2008. [Online]. Available : <http://hal.inria.fr/inria-00345669>