Inférence des poids d'un graphe des conflits modélisant un réseau Wi-Fi

Lafdal Abdelwedoud[1] et Anthony Busson[1] et Isabelle Guérin Lassous[1]

[1] *Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France.*

Les réseaux Wi-Fi en mode infrastructure sont composés de plusieurs points d'accès (AP). Ceux-ci, quand ils utilisent le même canal, interférent les uns avec les autres. Le graphe des conflits représente alors les interférences entre les AP. Les interférences ayant des impacts différents suivant leur niveau, nous considérons un graphe pondéré. Dans cet article, nous nous intéressons à l'inférence des poids de ce graphe. Notre méthode se base sur des mesures du taux d'activité du medium réalisées par chaque AP. Ces mesures sont récupérables en pratique. Nous montrons que notre méthode basée sur la décomposition du graphe et sur une approche de réseau Markovien permet d'inférer très précisément ces poids. La méthode est validée au travers d'un jeu de simulation sur le simulateur réseau ns-3.

Mots-clefs : Graphe des conflits, Wi-Fi, inférence.

1 Introduction

Les réseaux Wi-Fi en mode infrastructure sont aujourd'hui le premier moyen pour accéder à Internet. Sur les campus, dans les entreprises, ou dans certains lieux publics, un ensemble de points d'accès est déployé afin de couvrir la zone d'intérêt. On parle alors de ESS (Extended Service Set). Sur la plupart des produits commerciaux, les points d'accès de l'ESS sont contrôlés par un serveur central. Cela facilite l'administration et permet aussi une optimisation globale du réseau (configuration des canaux, associations, etc.). Cette optimisation doit s'appuyer sur des informations ou des mesures provenant des points d'accès (AP).

L'un des paramètres qui a une influence importante sur les performances du réseau Wi-Fi est **le graphe des conflits**. Dans le graphe des conflits, les nœuds sont les AP et les liens les interférences/conflits entre les AP. Un conflit existe si deux AP utilisent un même canal et qu'ils sont à portée de détection l'un de l'autre. Cette notion de détection est liée aux protocoles d'accès au medium et au mécanisme CCA (Clear Channel Assessment) qui indique si le medium est libre ou occupé à un instant donné. Si deux AP sont dans la zone de détection de l'un de l'autre alors ils détecteront leurs transmissions et ne pourront pas transmettre en même temps (à quelques collisions près). Cette détection CCA s'appuie sur plusieurs sous algorithmes : i) détection d'un signal reconnu comme un signal IEEE 802.11, ii) détection d'un niveau d'énergie sur le canal supérieur à un certain seuil, iii) une combinaison de i) et ii). Le fait pour des AP de se détecter a un impact fort sur les performances puisque cela signifie qu'en pratique ils doivent se partager le medium. Sur les normes récentes, IEEE 802.11n/ac/ad en particulier, les seuils sur le niveau d'énergie font que cette zone CCA est proche de la portée de communication radio (portée où l'on est capable d'interpréter les trames avec la modulation la plus robuste).

Nous avons réalisé une expérience dans notre laboratoire dans laquelle deux AP Wi-Fi sont positionnés à différentes distances et l'un des AP transmet en permanence du trafic. On observe alors le *busy time* sur le second AP. Le *busy time* est la proportion de temps où le nœud détecte le medium occupé comme indiqué par son mécanisme CCA. La figure 1 montre que celui-ci décroit avec la distance. La détection du trafic de l'AP voisin n'est donc pas en tout ou rien. Pour certaines distances seule une portion du trafic est détectée. Par exemple, un voisin transmettant x pourcent

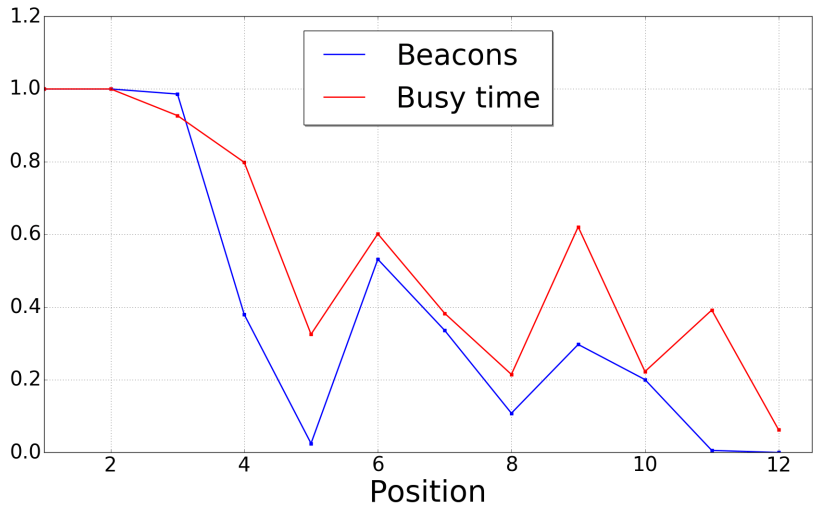


Figure 1: *Busy time* et taux de *beacons* reçus pour différentes distances.

du temps sera détecté seulement $r \cdot x$ pourcent du temps (avec $r \in [0,1]$). **Le graphe des conflits représentant les détections CCA est donc pondéré.**

Les AP transmettent périodiquement des *beacons*. Le pourcentage de *beacons* reçus pourrait nous permettre de fixer ces pondérations. Néanmoins ce pourcentage décroit plus vite que le pourcentage de trafic détecté par le CCA car une partie des *beacons* arrivent erronés. Il peut donc y avoir un décalage important entre le taux de réception des *beacons* et le CCA comme on l'observe sur la figure 1.

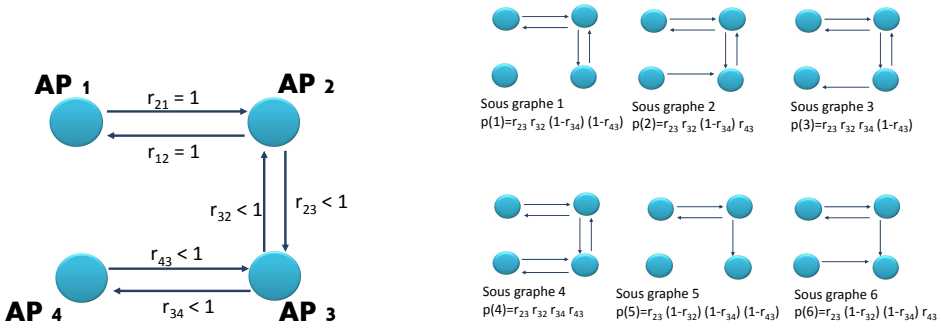
Afin d'estimer plus finement les pondérations du graphe des conflits, nous proposons une méthode passive (qui ne nécessite aucun trafic supplémentaire). Les expérimentations ont montré que lorsque le taux de réception des beacons était de 0 le taux de détection CCA l'était également. Même pour des taux de détection CCA très faibles, le récepteur a toujours été capable d'interpréter un beacon, parfois un sur mille, mais c'est suffisant pour déterminer l'existence d'une arête dans le graphe des conflits. Nous supposons donc que le graphe des conflits est connu mais pas les pondérations. Le contrôleur récupére les busy time des différents AP de son ESS (busy time mesurés sur une même période) et le trafic généré par chaque AP. Le *busy time* local mesuré par un AP dépend directement du trafic des autres AP et du taux de détection CCA (le poids que nous recherchons). Nous nous proposons donc d'inférer le graphe des conflits à partir de ces seules données. La méthode est décrite dans la prochaine section. Ces méthodes sont plus légères que les méthodes proposées dans la litérature [LZZ17, KPD10, PKMD13] qui s'appuient sur des mesures qui peuvent être, en pratique, compliquées à obtenir ou qui n'infèrent pas les poids mais le graphe en lui-même.

2 Modèle

2.1 Méthode d'inférence des poids

Nous considérons un graphe pondéré avec N AP/nœuds où une arête entre les nœuds i et j a un poids noté r_{ij} avec $0 < r_{ij} \leq 1$. Le graphe est orienté. r_{ij} représente la proportion de trafic provenant du nœud j et détecté par le nœud i. Le graphe est suppposé connu mais pas les poids sauf ceux égaux à 1.

Nous supposons connus les *busy time* de chaque AP mesurés sur une même période. Le vecteur correspondant est noté $\hat{B} = (\hat{b}_i)_{i=1,..,N}$. Le taux d'occupation du medium généré par le trafic local d'un AP durant la même période est noté x_i avec $x_i \leq \hat{b}_i$.



(a) Graphe des conflits

(b) Décomposition en sous-graphes

Figure 2: Le graphe des conflits (figure (a)) est décomposé en tous les sous-graphes orientés possibles (figures (b)). Ici, seuls les 6 premiers sont représentés (sur les 16 possibles).

L'idée est de proposer un modèle avec les activités x_i et les poids r_{ij} en entrée et d'en déduire les *busy time* théoriques notés $B = (b_i)_{i=1,..,N}$. Nous choisissons alors les r_{ij} qui minimisent l'erreur :

$$\min_{r_{ij},(i,j)\in\{1,..N\}^2} \|\hat{B} - B\|_2$$

où $\|.\|_2$ est la norme L^2.

2.2 *Busy time théorique - calcul de B*

Nous décomposons le graphe des conflits initial en tous les sous-graphes possibles avec la contrainte que les arêtes de poids 1 soient toujours présentes. Par exemple, pour la topologie considérée dans l'évaluation, nous répresentons en figure 2 le graphe des conflits original et sa décomposition en sous-graphes.

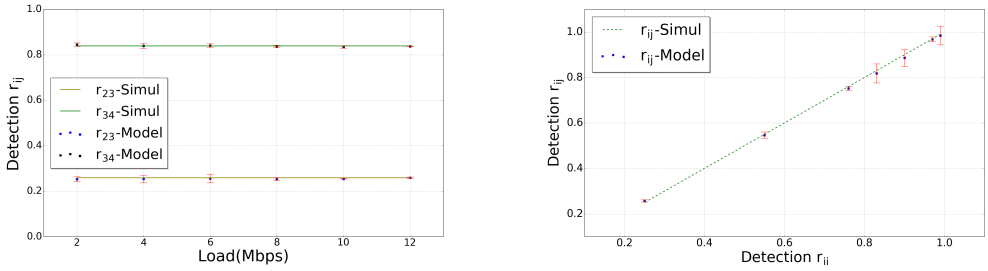
La probabilité d'occurence d'un sous graphe est le produit des poids de ces arêtes. La probabilité d'occurence du graphe l est noté $p(l)$. Par exemple pour le sous-graphe 2 de la figure 2 (b), sa probabilité est $p(2) = r_{23}r_{32}r_{43}(1 - r_{34})$. Pour un sous-graphe donné, nous supposons que toutes les arêtes ont un poids de 1. On peut voir chaque sous-graphe comme un des états de détections possibles entre les AP. Les sous-graphes n'étant pas pondérés, nous pouvons appliquer la méthode proposée dans [ABL18] qui calcule le *busy time* pour un graphe des conflits et une activité donnés. Pour le sous-graphe l, le *busy time* théorique de l'AP i est noté b_i^l. Soit SG l'ensemble des sous-graphes, le *busy time* théorique de l'AP i est alors calculé comme :

$$b_i = \sum_{l\in SG} p(l) b_i^l \tag{1}$$

Il est important de noter que les *busy time* des sous-graphes (b_i^l) ne sont calculés qu'une seule fois. L'optimisation se fait en pratique sur les coefficients $p(l)$.

3 Résultats numériques

Afin de valider notre approche, nous avons simulé dans ns-3 la topologie donnée figure 2 (a). Un *fading* de loi normale et un bruit plus important ont été rajoutés à la couche physique de ns-3 afin d'avoir un comportement similaire à celui décrit en figure 1. Des simulations préalables nous donnent les valeurs réelles des quatre inconnues que nous recherchons (r_{23}, r_{32}, r_{34} et r_{43}). Pour ce scénario, les liens sont symétriques ($r_{23} = r_{32}$ et $r_{34} = r_{43}$). Le problème d'optimisation est résolu



(a) La charge de trafic augmente.

(b) La valeur de r_{12} augmente.

Figure 3: Figure (a) : inférence des poids en fonction de la charge. Figure (c) : r_{23} augmente.

ici grâce à la méthode des moindres carrés implémentée sous matlab. Celle-ci suppose une valeur initiale des poids que nous tirons au hasard dans l'intervalle $[0, 1]$.

La figure 3 indique les résultats pour différents niveaux de charge variant de 2 à 12 Mbit/s. L'idée est de voir l'influence de la charge sur la précision de l'inférence. 5 simulations sont effectuées pour chaque niveau de charge. Pour chacune d'elle, nous obtenons donc une mesure des *busy time* réels et des taux d'activité locaux sur lesquels nous appliquons notre méthode. Le poids inféré est la moyenne des résultats obtenus sur les 5 simulations. Les résultats sont montrés avec un intervalle de confiance à 95%. Nous observons que les poids inférés sont très proches des poids réels et que la méthode est clairement insensible à la charge de trafic. Sur la figure 3 (c) nous avons éloigné l'AP 4 de l'AP 3 de manière à augmenter la valeur de r_{34} et voir l'impact de la valeur de r_{ij} sur la méthode. On observe clairement que la méthode fonctionne quelles que soient les valeurs de r_{ij}. Cependant l'intervalle de confiance augmente avec r_{34} montrant que les écarts des résultats d'une expérience à l'autre sont plus importants quand r_{34} est grand.

4 Conclusion

Nous avons proposé une méthode d'inférence des poids du graphe des conflits dans les réseaux IEEE 802.11 en mode infrastructure. La méthode s'applique aux amendements récents de la norme (IEEE 802.11n/ac/ad typiquement) où la portée radio se confond avec la zone de détection. Le graphe en lui-même est alors supposé connu. Pour une topologie simple, la méthode présente des résultats très précis lorsque celle-ci est comparée aux poids obtenus avec le simulateur ns-3. Des simulations sont en cours sur des topologies plus complexes et sur des liens asymétriques. La fonction à optimiser a été montrée convexe et nous nous appuyons sur d'autres méthodes d'optimisation que celle utilisée ici. Des tests sur la plateforme FIT IoT Lab sont également envisagés.

Références

[ABL18] Mohammed Amer, Anthony Busson, and Isabelle Guérin Lassous. Association optimization in wi-fi networks based on the channel busy time estimation. In *IFIP Networking*, 2018.

[KPD10] A. Kashyap, U. Paul, and S. R. Das. Deconstructing interference relations in wifi networks. In *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9, June 2010.

[LZZ17] W. Li, J. Zhang, and Y. Zhao. Conflict graph embedding for wireless network optimization. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.

[PKMD13] U. Paul, A. Kashyap, R. Maheshwari, and S. R. Das. Passive measurement of interference in wifi networks with application in misbehavior detection. *IEEE Transactions on Mobile Computing*, 12(3) :434–446, March 2013.