A simple method to infer Wi-Fi conflict graph

Anthony Busson, Eric Fleury, Ngoc Minh Phung

A simple method to infer Wi-Fi conflict graph

Anthony Busson[1] et Éric Fleury[1] et Ngoc Minh Phung,[2,1]

[1] *Univ Lyon, CNRS, ENS de Lyon, Inria, UCB Lyon 1, IXXI, LIP, 69342 Lyon FRANCE*
[2] *University of Sciences and Technology of Hanoi (USTH), Vietnam*

In Wi-Fi networks, CSMA/CA ensures that access points (AP) in conflict with each others cannot transmit at the same time. An AP detects a conflicting AP when the received signal strength is greater than a certain threshold. This mechanism sets the medium spatial reuse that strongly impacts the throughput that may be offered by the Wi-Fi network and the users quality of experience. The knowledge of the different conflicts is thus crucial if we want to optimize the Wi-Fi network. In this paper we propose to take benefit of all local statistics information gathered by APs wireless interfaces to reconstruct the global conflict graph, i.e. the graph that represents AP in conflict with each others. Our methodology is based on APs statistic information available in profile counters. Consequently, our method does not rely on complex captures or synchronization. The proposed method is a work in progress for which we propose a proof of concept through a first set of simulations.

Mots-clefs : Wi-Fi, interference, ILP, conflict graph

1 Introduction

Wi-Fi wireless local area networks (WLAN) may be composed of several access points (AP) using the same service set identification (SSID). This service is commonly known as an extended service set (ESS). ESS offers internal roaming: a mobile station may move and be associated transparently from one AP to another. In order to ensure these transparent handovers, but also a good coverage and high transmission rates (modulation and coding) we observe a densification of APs. At a given time, a mobile station may have several APs of the same ESS in its transmission range. The Wi-Fi network parameterization in terms of associations or channels assignment is consequently not trivial. Nevertheless, these decisions are crucial if one want to optimize the WLAN in terms of resource usage, throughput, load balancing (stations distributed homogeneously between APs). This parameterization strongly impacts the throughput that may be offered by the Wi-Fi networks and the users quality of experience. Nowadays, such decisions tend to be centralized, through AP controllers which manage a set of *thin* APs. But, an AP controller requires information and statistics about AP, channels usage, number of stations, etc. to take its decisions. Obviously, one of the most important information needed is the conflict graph between APs. Indeed, associating a mobile station to an AP will impact this AP in terms of bandwidth but also all other APs that are able to detect the signal from this communication. Unfortunately, the conflict graph is generally unknown. APs can list APs that are in their transmission range, since they are detected from their beacons, but they are not able to identify APs for which the signal is undecodable.

Several studies deal with this problem. A simple way to infer the conflict graphs is to inject traffic at each AP [Nic07, AK06]. APs detecting a busy medium during the traffic injection are then assumed in conflict with the transmitting one. Obviously, passive methods are less intrusive and have the benefit to be performed at anytime without disturbing communications in progress. In [PKMD13, KPD10, YDN16], traffic captures are performed on wireless interfaces by each AP and correlated. Correlations are linked to the CSMA/CA mechanism where interfering APs do not transmit at the same time except when collisions occur. The conflict graph is, for a part, inferred from collisions detection. The two APs victim of a common collision are assumed in conflict. These methods are quite complex as a capture that analyzes traffic and collisions has to be performed. It requires very specific tools and a synchronization between APs to match a

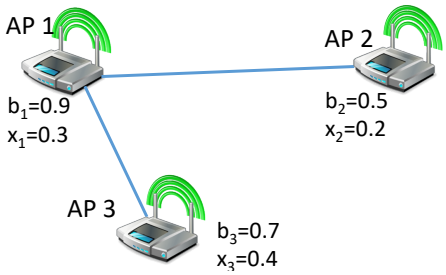

Fig. 1: A simple example with three AP. Lines represent conflicts between APs.

collision to a set of frame transmission failures on several APs. In [CBFK09], traffic from APs is analyzed on the distributed system (the wired network) at the router. It correlates traffic from the different APs: two interfering APs do not send data at the same time and thus when one AP is transmitting more than the other one, the other one has a traffic decrease, and vice versa.

In this paper we propose to take benefit of all local statistics information gathered by APs wireless interface such as the amount of tx/rx bytes or the channel occupancy in order to reconstruct the global conflict graph. From local AP information, we derive for each oriented pair (u,v) of APs the influence in term of medium occupancy of u on v. The main goal is to derive the global matrix of conflict between all pairs of APs, that is, to be able to uncover all direct conflicts when two APs are in their transmission range (*communication possible*) but also conflicts when station are in their detection range (*detection of the signal but no communication possible*). It is important to note that we base our methodology on APs statistic information available in profile counters. Consequently, our method does not rely on complex captures or synchronization. Each AP sends these values to a central controller which obtain the statistics for all APs of the ESS. Applying our method, it deduces the conflict graph. The proposed method is a work in progress for which we propose a proof of concept through a first set of simulations and scenarios.

2 Model and formalism

Formalism Let assume that the ESS network for a given channel is composed of n connected access points $V = \{v_1, \ldots, v_n\}$. For each access point $v_i \in V$ we collect the following statistic information:

- x_i is the proportion of time the station v_i is transmitting on the wireless medium.
- b_i is the proportion of time the station v_i senses the medium as busy.

Of course, the local busy time b_i, includes the time spent by the station itself to send its own data: $b_i \geq x_i$. It also includes the time spent when it received data from other base stations (in its transmission range), and the time spent when the AP senses the medium as busy without being able to decode it. For the latter, the signal is not strong enough to be properly received and decoded but strong enough to forbid any other reception or emission. Therefore, we have an incomplete puzzle to solve in order to uncover the global conflict matrix denoted $A = (a_{i,j})_{1 \leq i,j \leq n}$ in the following. Indeed, it is straightforward to fill conflicts between a base station v_i and v_j if v_i receives messages from v_j (v_i is in the transmission range of v_j and it receives at least the beacons of v_j). These edges being known, we fill a first part of the conflict matrix $A = (a_{i,j})_{1 \leq i,j \leq n}$ and set the corresponding values $a_{i,j} = 1$. The other terms are unknown and are left unset. The goal is now to uncover other conflicts between stations v_i and v_j when v_i is in the detection range of v_j but not in the transmission range. These conflicts are inferred through an optimization problem for which the parameters are the measured statistics $(x_i)_i$ and $(b_i)_i$.

A simple example Before describing the method in a formal way, we present the problem and the proposed method with a simple example. We consider three APs as shown in Figure 1. The links represent

Objective functions	Number of APs					
	5	6	7	8	9	10
Maximizing the number of links	95.8	91.8	83.1	70.1	50.9	38.9
Minimizing errors	100	100	100	100	100	100

Tab. 1: Percentage of correctness with the two objective functions.

Errors (%)	2%	5%	10%	20%	50%
1 run	98%	91%	78%	73%	65%
10 runs	100%	99%	91%	77%	70%
20 runs	100%	100%	93%	81%	75%

Tab. 2: Percentage of correctness in the presence of errors.

the conflicts. In this example, APs are not in the transmission ranges of each others. The conflicting links correspond to signals that are detected by a neighbor but not decoded. Each AP is able to measure its own activity (x_i) and the busy time (b_i). These values are sent to the AP controller. The controller tries to find the conflict graph for which each busy time is the sum of the activity times of the APs in conflict. For instance, AP 2 cannot be in conflict with both APs 1 and 3 as it would lead to a busy time of 0.9 ($x_1 + x_2 + x_3 = 0.9$) and a busy time of only 0.5 is measured. AP 2 can be in conflict with AP 1 but not with AP 3 as its busy time $b_2 = 0.5$ is less than $x_2 + x_3 = 0.6$. The inferred conflict graph is the one that verifies all these constraints.

The method The underlying idea consists in writing the constraints on the b_i and x_i with respect to the conflict graph and solve the reverse problem. In other terms, we find the conflict graph for which the constraints are verified. The constraints are as follows:

$$\begin{cases} \sum_{i,j}^{n} a_{i,j} x_j \leq b_i \text{ with } a_{i,j} \text{ equals 1 or 0} \\ a_{i,j} = 1 \text{ if AP } i \text{ receives beacons from AP } j \\ a_{i,j} = 1 \text{ if } a_{j,i} = 1 \\ a_{i,i} = 1 \end{cases} \tag{1}$$

The first constraint expresses that the proportion of time the AP v_i is sensing the medium busy must be greater than the sum of all time used by all station v_j in conflict with it. The second constraint expresses that the conflict graph is symmetric. An AP that detects transmissions from another AP has its own transmissions detected by this AP. The third constraint expresses that the time an AP is spending to send its own data accounts for the global busy time.

The solution is found through the optimization of two different objective functions. For the first function, we maximize the number of links in conflict: $\max\left(\sum_{i,j}^{n} a_{i,j}\right)$.

A second objective function reflects the error between the measured busy time and the sum of the activity times x_i of a given conflict graph. It aims to be minimized: $\min\left(|(b - A \cdot x)|\right)$.

3 Numerical results

We evaluate our approach through a set of simulations. These simulations have been run on matlab. We consider topologies with a number of APs ranging from 5 to 10. APs are independently and uniformly distributed in a window with size 800 meters $\times$ 400 meters. Radio and detection ranges are set to 120 and 280 meters respectively. Consequently, we obtain samples of topologies for which the communication and the conflict graphs are calculable. We associate to each AP i an activity x_i which is randomly drawn in $[0, 1]$. The proportion of time the medium is sensed busy, b_i, is deduced from the set $(x_i)_i$ and the conflict graph. For a given number of APs, we generate 50 different topologies. We apply the proposed method to infer the conflict graph considering the two objective functions. Optimization is obtained through the MILP(Mixed Integer-Linear programming) solver of matlab. Results are shown in Table 1. For each simulation, we compute the proportion of matching links between the real conflict graph and the one obtained through the optimization. The presented percentages are the average of these proportions over the 50 topologies.

The objective function maximizing the number of edges in the conflict graph offers accurate results for 5 and 6 APs but its accuracy decreases to a modest 40 % for 10 APs. Beside, minimizing the errors lead systematically to a perfect match of the conflict graph whatever the number of APs. As the measures are perfects, i.e. the values b_i correspond exactly to the sum of x_i according to the conflict graph, the error reaches 0 once the solution has been found. When the number of conflicts is maximized, several solutions may exist and the one with the highest number of edges is chosen even if the error is greater. With an increasing number of APs, the possibility to find alternative conflict graphs with regard to the real one increases which explained the poor performance of this objective function.

The parameters x_i are based on the APs activity. It implicitly assumes that we take into account only download traffic (which composes, in practice, the main part of the traffic load). Instead, it is also possible to integrate upload traffic in these parameters. In both cases, it introduces an error between the real b_i measured by access points and the sum of the activity for the APs in conflict. In Table 2, we show the percentage of correctness for different levels of errors ("1 run"). In order to improve these results, we propose a method where the AP controller collects several measures, taken at different times from the AP, and run our algorithm as many times. Then, it considers that a conflicting link exists between two APs only if it appears a majority of times among the different runs. Table 2 reports the results for 10 and 20 runs. The method clearly improves the correctness.

4 Conclusion

We proposed a simple method to infer interference conflict graph in a Wi-Fi network. The originality of our method is to propose an algorithm which relies only on AP local counters and consequently does not require probe traffic, complex capture on the wireless medium, nor precise synchronization between APs. A first set of simulations shows that the conflict graph that minimizes the error between the local measures correspond systematically to the real one. This work is a preliminary work. The method has to be extended to take into account real deployment of Wi-Fi networks, for instance considering external Wi-Fi networks (not in the considered ESS) for which such measures are not available. Experimentations on the R2LAB/FIT platform [†] is in progress.

References

[AK06] N. Ahmed and S. Keshav. Smarta: A self-managing architecture for thin access points. In *Proceedings of the 2006 ACM CoNEXT Conference*, pages 9:1–9:12, 2006.

[CBFK09] Kan Cai, Michael Blackstock, Michael J. Feeley, and Charles Krasic. Non-intrusive, dynamic interference detection for 802.11 networks. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, 2009.

[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)*, June 2010.

[Nic07] Dragoş Niculescu. Interference map for 802.11 networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 339–350, 2007.

[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.

[YDN16] J. Yang, S. C. Draper, and R. Nowak. Learning the interference graph of a wireless network. *IEEE Transactions on Signal and Information Processing over Networks*, PP(99):1–1, 2016.

[†] https://r2lab.inria.fr/