



A Passive Method to Infer the Weighted Conflict Graph of an IEEE 802.11 Network

Lafdal Abdelwedoud, Anthony Busson, Isabelle Guérin-Lassous, Marion Foare

► To cite this version:

Lafdal Abdelwedoud, Anthony Busson, Isabelle Guérin-Lassous, Marion Foare. A Passive Method to Infer the Weighted Conflict Graph of an IEEE 802.11 Network. AdHoc-Now 2019 - 18th International Conference on Ad Hoc Networks and Wireless, Oct 2019, Luxembourg, Luxembourg. pp.304-316, 10.1007/978-3-030-31831-4_21 . hal-02404943

HAL Id: hal-02404943

<https://inria.hal.science/hal-02404943>

Submitted on 11 Dec 2019

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.

A Passive Method to Infer the Weighted Conflict Graph of an IEEE 802.11 Network

Lafdal Abdewedoud¹, Anthony Busson¹[0000–0002–9445–637X], Isabelle Guérin-Lassous¹[0000–0002–9558–5536], and Marion Foare^{1,2}[0000–0002–5404–950X]

¹ Univ Lyon, UCBL, EnsL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France
Firstname.Name@univ-lyon1.fr

² CPE, France
Firstname.Name@cpe.fr

Abstract. Wi-Fi networks often consist of several Access Points (APs) to form an Extended Service Set. These APs may interfere with each other as soon as they use the same channel or overlapping channels. A classical model to describe interference is the conflict graph. As the interference level varies in the network and in time, we consider a weighted conflict graph. In this paper, we propose a method to infer the weights of the conflict graph of a Wi-Fi network. Weights represent the proportion of activity from a neighbor detected by the Clear Channel Assessment mechanism. Our method relies on a theoretical model based on Markov networks applied to a decomposition of the original conflict graph. The input of our solution is the activity measured at each AP, measurements available in practice. The proposed method is validated through ns-3 simulations performed for different scenarios. Results show that our solution is able to accurately estimate the weights of the conflict graph.

Keywords: Wi-Fi · Weighted conflict graph · Inference

1 Introduction

Wi-Fi networks in infrastructure mode are nowadays the most used technology to access the Internet. In public areas, campuses, or companies, Access Points (APs) are deployed in order to cover the areas of interest. A Wi-Fi network often consists of several APs and forms an Extended Service Set (named ESS). In order to simplify the management of an ESS, a centralized approach is often considered. Proprietary or standardized solutions [4] allow the network administrator to control the ESS through a centralized controller. The controller offers a single interface to manage the ESS but also helps to optimize network resources via channel allocation, user association [2], identifying configuration issues (like hidden terminal for instance), etc. These optimizations may significantly increase the network performance like, for instance, throughput, fairness, and eventually the users' quality of experience.

In this context, an efficient network configuration relies on a deep and accurate knowledge of the current state of the Wi-Fi network. The parameters of

interest can be the channels used on APs, the users' association with APs, the profile of traffic transmitted in the network, APs' load, etc. One key parameter, related to the network performance, is the **conflict graph**. The conflict graph is a model capturing the conflicts between devices in the Wi-Fi network. For instance, when two devices use the same channel, or overlapping channels, they may detect and/or interfere with each other. The nodes in the conflict graph represent the devices of the network and there is a link between two interfering devices in the conflict graph. The conflict graph is very useful to manage Wi-Fi networks (for instance to allocate channels, to choose the channel width, etc.) and to predict the network performance.

In the IEEE 802.11 standard, the signal detection is performed through the clear channel assessment (CCA) mechanism that indicates whether the radio medium is busy or idle. If two nodes do not detect each other, they may potentially transmit at the same time, otherwise they have to share the medium and transmit at different times. In recent IEEE 802.11 standards, in particular in the IEEE 802.11n/ac amendments, the CCA detection threshold is sufficiently high to ensure a proper reception of the frames, at least for the most robust Modulation and Coding Scheme (MCS). In this case, the detection area corresponds to the radio range. The radio range refers here to the area where frames may be correctly received when the most robust MCS is used. Two nodes in detection range (or radio range) are called neighbor nodes hereafter.

The CCA does not systematically detect the whole activity of a neighboring node. As we will show in this paper, when a node is at the edge of a detection area, only a part of its transmissions may be detected. In this case, the medium is not totally shared but **partially shared** between the two nodes. The classical conflict graph does not model such a phenomenon since a link in the conflict graph indicates that the two endpoint nodes are in conflict all the time. We think that a **weighted conflict graph** is more appropriate to represent the level of detection/conflict between nodes.

The notion of weighted conflict graph is not new, but, in this paper, we propose an original method to build, from any Wi-Fi network, its weighted conflict graph. More specifically, we design a method to infer the weight of the conflict graph based on measurements available in practice on most of the commercial products. Unlike previous solutions, our method does not rely on distances between nodes. The measurements concern the local activity and the busy time fraction. The local activity is the proportion of time a given node uses the channel for its own transmissions and receptions. The busy time fraction of a given node is the proportion of time this node detects the medium busy according to the CCA mechanism. Our method is based on a Markov network model that gives the theoretical busy time fraction for a given activity and a given weighted conflict graph. Then, the method searches the best weighted conflict graph minimizing the error between the measured busy times and the theoretical ones. Our approach is validated through a set of simulations performed with the network simulator ns-3.

The paper is organized as follows. In the next section, we present the problem statement and the state-of-the-art. The model is described in Section 3. The validation is performed through ns-3 simulations in Section 4. We conclude in the last section.

2 Problem statement and state-of-the-art

2.1 Problem statement

The conflict graph is a key parameter for radio networks as it is used to model the potential conflicts in terms of radio medium sharing. Many studies base their solutions on a conflict graph. This latter is very often considered as an input of the problem and most of the studies do not explain how to build this conflict graph from a given network topology.

In Wi-Fi networks, the radio medium sharing is ruled by the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) principle. Two nodes share the radio medium when they are in detection range of each other and can not transmit at the same time except when their backoff simultaneously reaches zero. The random draw of the backoff limits these possibilities. Two nodes, that are not in detection range but whose transmissions interfere, are also considered as sharing the radio medium as their parallel transmissions may not lead to successful transmissions.

In this paper, our aim is to build the conflict graph that models the medium sharing due to the detection of the neighboring nodes' activity. We assume that the RTS/CTS mechanism is disabled. The activity detection by a Wi-Fi node is provided by the CCA mechanism. With the recent versions of IEEE 802.11, like 802.11n/ac, CCA assesses the medium occupancy according to three modes: 1) if the energy on the channel is greater than a given detection threshold 2) if a compliant IEEE 802.11 signal is detected 3) or if a combination of the first two modes appears. For most of the first Wi-Fi products based on the IEEE 802.11 standard of 1999, the detection range was almost two times the communication range induced by a use of a physical transmission rate of 2 Mb/s [7, 8]. The CCA threshold for the IEEE 802.11n/ac versions has been raised compared with the first amendments. In the IEEE 802.11 standard version of 2016, the medium is considered as busy by the CCA mechanism if “the start of a valid OFDM transmission at a receive level greater than or equal to the minimum modulation and coding scheme sensitivity” or if “a received energy that is 20 dB above the minimum modulation and coding scheme sensitivity” is detected. It means that the detection range corresponds to, at most, the communication range when the most robust MCS is used.

With such a rule, one might think that the conflict graph can be simply deduced from the neighborhood graph. Indeed, two nodes can detect each other as soon as they can communicate by using the most robust MCS. To illustrate that this is not so simple, we conducted a simple real experiment. We set up a first IEEE 802.11n AP, named AP1, that is always transmitting to a near

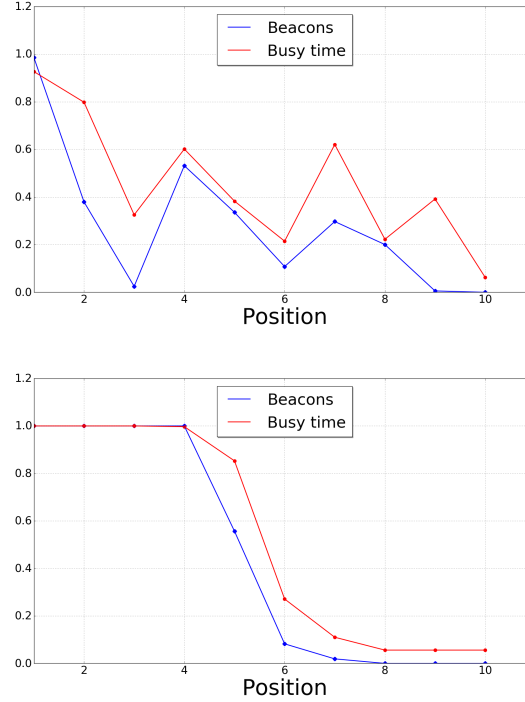


Fig. 1. Experimentations (up): busy time fraction and beacon reception rate for different distances. The different positions correspond to different distances given in an increasing order. Simulation ns-3 (down): we change the parameter of the ns-3 simulator to have a behavior close to the one obtained through experimentations. A log-normal fading has been added to the log distance propagation model and the level of noise has been increased.

station. The physical transmission rate is set by the default Wi-Fi manager. AP1 also periodically emits beacon frames to announce its network and some of its parameters. These beacon frames are sent with the most robust MCS at 6 Mbit/s. A second AP, named AP2, is located at different distances from AP1. AP2 measures two parameters: the busy time fraction and the percentage of received beacons. The obtained values are shown in Figure 2.1. When AP2 is in the detection area and close to AP1, the measured busy time fraction is close to 1. When the distance increases between AP1 and AP2, the busy time fraction decreases and evolves between 0.8 and 0.2. When AP2 is far from AP1 and outside the detection area (position not shown in Figure 2.1), the busy time fraction reaches 0. This experiment shows that, as soon as AP2 is not close to AP1, the CCA mechanism detects only a proportion of the transmissions but not all transmissions. It means that even if the two APs are in conflict, the medium is not totally exclusive and can be used, sometimes, at the same time by the two

APs. Therefore, the links in the conflict graph are not binary. Two nodes can be sometimes in conflict and sometimes not. To model this property, we suggest to use a weighted conflict graph in which weights on links represent the partial radio medium sharing between neighbor nodes.

We initially thought that the percentage of received beacons could be used to estimate the link weights of the conflict graph. Nevertheless, with our simple experiment, we observe that the percentage of received beacons may be lower than the busy time fraction measured by the CCA mechanism. This is due to the fact that some beacons arrive in error, even if they are detected by the CCA mechanism. Also, we observe that as soon as the CCA detection is not 0, the proportion of received beacons is strictly positive. It may be one beacon over 1000 but it is always positive.

According to these different observations, we set up the conflict graph in the following way:

- The nodes of the conflict graph are the APs of the considered Wi-Fi network.
- An edge exists between two nodes of the conflict graph if one of the corresponding APs is able to at least detect one beacon of the other AP.
- We associate to each link (a, b) a weight $w_{a,b}$ corresponding to the busy time fraction due to the activity of node b detected via the CCA mechanism by node a . A weight of 1 means that the medium is always detected as busy due to the activity of the neighbor node.
- As radio links are generally not symmetric, the conflict graph is directional and a different weight may be assigned in each direction of a link ($w_{a,b} \neq w_{b,a}$).

Note that $w_{a,b}$ may be also interpreted as the probability for a to detect a transmission from b through its CCA mechanism. The main difficulty lies in the estimation of the weights of the conflict graph. Indeed, an AP, when communicating with its associated stations, do not always use the most robust MCS. On the contrary, it adapts the physical transmission rate to use in order to get the best throughput with each of its associated stations. When transmitting with a MCS corresponding to a fast transmission rate, a neighbor AP may decode the physical header of the frame sent with the most robust MCS and may not decode the MAC header and the payload of the frame sent with another MCS. In this case, the CCA mechanism will indicate to the node that the medium is busy for the intended duration of the transmitted frame (duration indicated in the frame physical header), but the node will be unable to know from which node the frame has been sent because it will be unable to decode the MAC (or the IP) address, and thus unable to infer the weight with this neighbor AP. Moreover, the busy time at a node cannot be computed as the sum of the activity of its neighbors. Indeed, these neighbors may not be in conflict and their transmission may overlap in time. The medium may then be detected as busy according to the CCA mode 1. In the following, we describe a passive method to infer the weights w of the conflict graph.

2.2 State-of-the-art

A large set of papers use a conflict graph to provision Wi-Fi networks or to estimate the Wi-Fi network performance. In most of these studies, the authors assume that the conflict graph is known and do not explain how to build this conflict graph (like for instance in [10]). Few papers are dedicated to the computation of the conflict graph. Moreover, most of these studies consider that nodes (or links), that are identified in conflict, are permanently in conflict, which is not true as shown in the previous section. This is for instance the case in [11, 13, 15].

In [3], the authors propose to investigate network traffic at wired routers that interconnect Wi-Fi networks to the Internet in order to infer which nodes interfere to each other and to which level. This method only concerns congested networks, whereas our solution can be applied whatever the load in the network. In [12], the authors also consider the notion of partial conflict. But, contrary to our approach that is focused on the neighbors' activity, this work aims at assessing the interference between nodes via received signal strength (RSS) measurements. Even if the RSS is a parameter of interest, it is difficult to infer, with this parameter, the performance, e.g. the throughput, of a Wi-Fi link as shown in [9]. In [14], the authors also propose a passive measurement framework to infer the neighbors' activity. Their approach requires to compare trace logs between any pair of nodes whereas our solution only needs to measure the busy time on each node. In [8, 6], the authors build a weighted conflict graph from a Wi-Fi network in which overlapping channels may be used. In both solutions, the link weight is based among other things, on the distance between the two endpoints. Finally, in [5], the authors describe a method to measure the interference level on a node or a link and its impact on the network performance, but they do not build a conflict graph with their measurements.

As far as we know, our solution is the first one to consider the CCA measurements to determine the weights of a conflict graph.

3 Method to infer the weighted conflict graph

3.1 Inputs of the proposed method

We consider a weighted directed graph $G = (V, E)$ where V is the set of vertices representing the APs with $|V| = N$ and E is the set of directed edges. A directed edge (j, i) exists if AP_i is able to detect at least one beacon of AP_j . In this case, the weight w_{ij} , taking its value in the interval $[0, 1]$, is associated to the directed edge (j, i) . An example of such a conflict graph is given in Figure 2.

The weighted conflict graph can also be expressed through its matrix form:

$$W = \begin{pmatrix} 1 & w_{12} & w_{13} & w_{14} \\ w_{21} & 1 & w_{23} & w_{24} \\ w_{31} & w_{32} & 1 & w_{34} \\ w_{41} & w_{42} & w_{43} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & w_{23} & 0 \\ 0 & w_{32} & 1 & w_{34} \\ 0 & 0 & w_{43} & 1 \end{pmatrix} \quad (1)$$

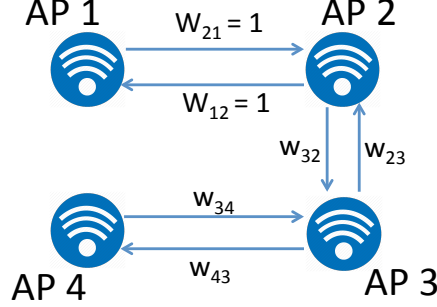


Fig. 2. The topology considered throughout this paper. AP_2 detects almost all beacons from AP_1 . They are assumed to be permanently in conflict and the weight w_{21} is set to 1. AP_2 detects only a part of the beacons from AP_3 . The weight is thus supposed unknown and denoted w_{23} . AP_2 does not receive any beacon from AP_4 , the weight w_{24} is then set to 0. The same principles are applied to AP_1 , AP_3 , and AP_4 .

In this matrix, each term w_{ij} is the weight of the directed edge (j, i) in the conflict graph. When AP_i does not receive any beacon from AP_j , the weight w_{ij} is set to 0 and there is no edge from AP_j to AP_i in the conflict graph.

If the proportion of received beacons is greater than a given threshold (close to 1), we set the weight to 1. It allows to reduce the number of unknown weights for a given topology.

The weights are inferred from the busy times. We assume that the busy times are measured on each AP during the same period. It corresponds to the proportion of time during which the medium is sensed busy by this AP according to its CCA mechanism. When the AP is transmitting, the medium is also assumed busy. The busy time measured on AP_i is denoted \bar{b}_i . The corresponding vector, denoted $\bar{B} = (\bar{b}_i)_{1 \leq i \leq N}$, gives the busy time measured by each AP of the network. The local activity (transmission/reception) of an AP_i is denoted x_i . It corresponds to its own contribution in terms of transmissions and receptions to \bar{b}_i ($x_i \leq \bar{b}_i$).

To compute the weights of the conflict graph, we can not rely on the identity of stations occupying the radio medium with their transmissions, because, as explained in Section 2, the measurement of the busy times does not always allow to know which station causes this medium occupancy. Instead, we propose, in the next section, a model that computes a theoretical busy time at each AP for a given set of weights. These theoretical busy times are denoted $B(W) = (b_i(W))_{1 \leq i \leq N}$ for a given vector of weights $W = (w_{ij})_{1 \leq i \leq N; 1 \leq j \leq N}$. The inferred weights are then the ones that minimize the difference between the theoretical and the measured busy times.

$$\widehat{W} = \arg \min_W \|B(W) - \bar{B}\|_2 \quad (2)$$

where $\|\cdot\|_2$ is the L^2 norm.

3.2 Theoretical busy time calculation

Our method relies on the solution proposed in [2]. In this solution, the authors estimate the theoretical busy time for each AP with a Markov network model knowing the conflict graph of the network and the activity of each AP. The used conflict graph is undirected and no weight is considered in this study. It only models the detection activity between APs, which is assumed to be known. We extend this work to estimate the theoretical busy times at each AP for a directed weighted graph.

First, we propose to decompose the weighted conflict graph, given as an input, in all possible sub-graphs. The set of these sub-graphs is denoted SG . The directed edges with a weight of 1 are always present in all the sub-graphs. For all the other edges whose weight is not null and less than 1, they can be present in each sub-graph or not. When such an edge is present in a sub-graph, a weight of 1 is then given to this edge.

We define a probability of occurrence $p(g, W)$ to each sub-graph $g \in SG$ defined as $g = (V, E_g)$ where E_g is the set of directed edges present in g . To compute this probability, we associate to each directed edge $(j, i) \in E$ the probability w_{ij} if the edge (j, i) is present in the sub-graph and $1 - w_{ij}$ otherwise. The probability of occurrence of the sub-graph g is then the product of these probabilities for all possible edges $(i, j) \in E$, as follows (1. is the indicator function):

$$p(g, W) = \prod_{(j,i) \in E} (w_{ij} 1_{(j,i) \in E_g} + (1 - w_{ij}) 1_{(j,i) \notin E_g}) \quad (3)$$

For our example given in Figure 2, some possible sub-graphs are described in Figure 3. The computation of $p(g, W)$ for the presented sub-graphs is also given.

In each sub-graph, all link weights are set to 1. It is then possible, for each sub-graph, to estimate the theoretical busy time of each AP knowing the APs' activities in this sub-graph. A variant of the method in [2] is applied to the sub-graph to obtain these theoretical busy times. The only difference of this variant with the original method is that it takes into account the directed links in the computation of the busy time. More precisely, the busy time computation at a node i counts only links directed to i (all the links (j, i) but not (i, j)). The theoretical busy time estimated for node i in the sub-graph g is denoted b_i^g .

Finally, the theoretical busy time of a node i in the initial weighted directed conflict graph (given as an input) is then given by:

$$b_i(W) = \sum_{g \in SG} b_i^g \cdot p(g, W) \quad (4)$$

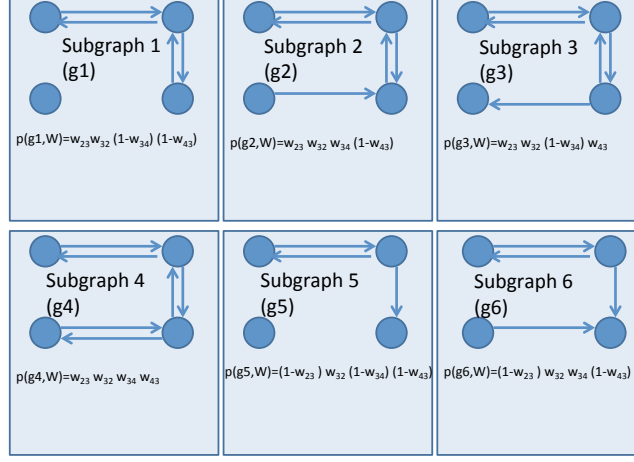


Fig. 3. Possible sub-graphs from the initial graph given in Fig. 2. The edges of weight 1 are present in all the sub-graphs (edges between AP_1 and AP_2). The number of possible sub-graphs for this example is then 2^4 (there are 4 unknown weights in the initial graph). Only 6 of these sub-graphs are represented with their probability of occurrence $p(g_i, W)$.

3.3 Weight computation

The method, presented in the previous section, computes the theoretical busy time of each AP for a weighted conflict graph for which the link weights are given. But the link weights are not known and they must be inferred. As mentioned in Section 3.1, the inferred weights are the ones that minimize the error between the theoretical and the measured busy times as given in Equation 2. To this end, we compute, for all the possible sets of weights, the theoretical busy times for all the APs and compare the estimated theoretical busy times with the measured ones. A step of 0.005 is considered to explore the weight space.

4 Numerical results

Our approach is validated through simulations performed with the network simulator version 3 ns-3 [1]. We consider two scenarios simulating an IEEE 802.11n Wi-Fi network: the one presented in Figure 2 where the weights are symmetric ($w_{23} = w_{32}$; $w_{34} = w_{43}$ and $w_{21} = w_{12} = 1$), and a second scenario with the same topology but with 6 weights to infer asymmetric links. The default path-loss function implemented in ns-3 has been modified to mimic the behavior observed in our experiments. It is described in bottom figure of Figure 1. In order to get asymmetric weights for the second scenario, we set different antenna

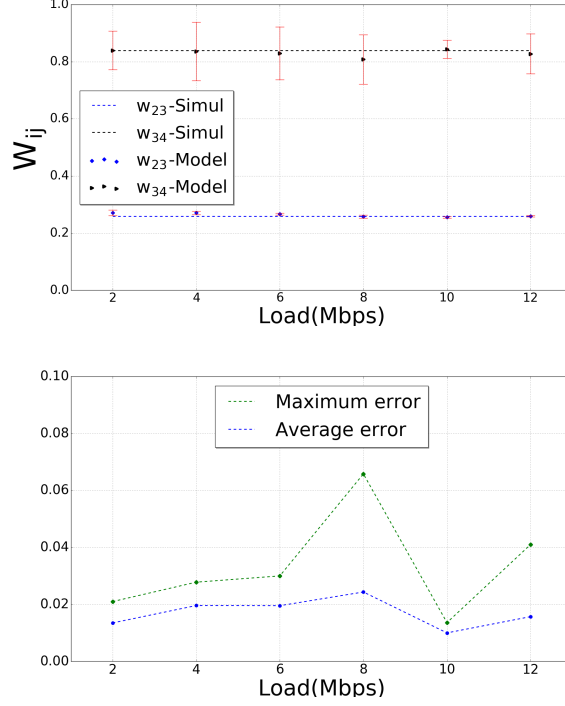


Fig. 4. Scenario 1 - comparison between the estimated and the real weights: values and errors. On top, the y-axis represents the values of the weights, $w_{23} = 0.25$ and $w_{34} = 0.83$. On the bottom figure, the error is computed as the absolute value between the inferred weight and the real weight. In the figure, we show the average and the maximum error for the 5 samples in each simulation.

gain to the nodes. We perform 5 ns-3 simulations for each configuration and for a given network load, i.e. for each point in the figures. For each simulation, we measure the activity and the busy time for each node. The optimization problem is then solved for this input. Each point is then the mean of the 5 simulations shown with a confidence interval at 95%. The obtained weights are compared to the real ones that can be easily obtained from the simulator. The size of the considered networks in this section allows us to find the global optimum through an exhaustive search, but any other approaches could be used instead, in particular for more complex scenarios.

4.1 Scenario 1: 4 nodes with symmetric links

In Figure 4, we show a comparison between the inferred and the real weights. Results show that our method is very accurate with a mean error approximately equals to 0.02 and a maximum error that does not exceed 0.07. In these two

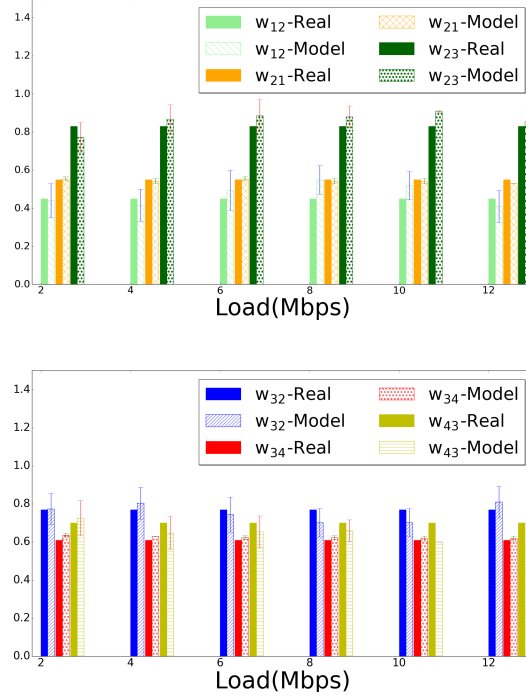


Fig. 5. Scenario 2 - comparison between the estimated and the real weights.

figures, we vary the load on the APs to observe its impact on our method. For this scenario, our approach is insensitive to the load and works for both unloaded and congested networks. 5 measures collected on the APs were sufficient to obtain accurate results. Nevertheless, we can observe on the obtained weights for each simulation and the confidence interval that a single measure does not always lead to accurate results. Instead, the computation of the mean of several samples is necessary to obtain more accurate weights. Therefore, the controller must regularly collect measures from the APs to refine its estimation.

4.2 Scenario 2: 4 nodes with asymmetric links

We consider a scenario with the same topology as the one described in Figure 2. There are two differences with the previous scenario: the weights w_{12} and w_{21} are not 1 and must be inferred and the links are not symmetric. The asymmetry is created through different gains associated to the Wi-Fi cards. The real weights that we want to infer are $w_{12} = 0.45$, $w_{21} = 0.55$, $w_{23} = 0.83$, $w_{32} = 0.77$, $w_{34} = 0.61$, and $w_{43} = 0.7$.

We compare the values obtained with our method with the real weights in Figure 5. The system load varies from 2 to 12 Mb/s. For this second scenario,

it appears that the asymmetry between the weights of a same link does not impact the accuracy of our method. It is thus able to give accurate values of the weights even if the graph is undirected. It is a crucial feature of our method as Wi-Fi networks naturally introduce different behaviors at each node due to the radio environment and the wireless cards properties (gain for instance). As it is suggested by the confidence interval, a unique measurement was not enough to accurately estimate the weights. In Figure 6, we show the errors (average and max) for the 5 simulations. The errors is not insensitive to this load. But, the mean error does not exceed 0.05 and the maximum error over the 5 samples is less than 0.1. It demonstrates that the controller must collect several measures from the APs at different times to obtain accurate estimations.

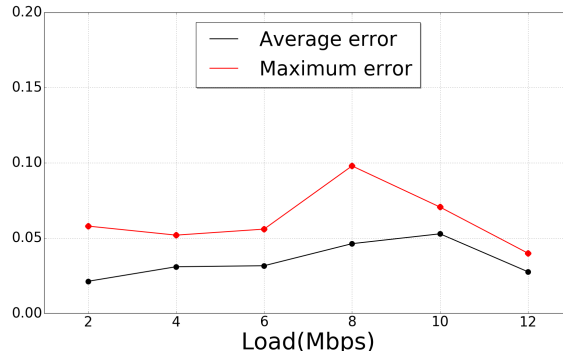


Fig. 6. Scenario 2 - The mean and maximum errors.

5 Conclusion

In this paper, we propose a passive method to infer the weights of a conflict graph modeling Wi-Fi networks based on recent IEEE 802.11 standards. The weight is defined as the proportion of the neighbor's activity detected by the Clear Channel Assessment mechanism. Our solution combines busy time measurements and the busy time estimations with an appropriate model. For the considered scenarios, the method has been shown very accurate and is able to infer the weights with approximately 2% of errors. Numerical results also suggest that the method is quite insensitive to the system load and the values of the weights. The numerical results show that it is necessary to collect AP measurements on different period of times to reach this level of errors. In a near future, we plan to validate the method through real experiments. Beside, the method has to be improved to take into account other sources, not belonging to the Extended Service Set.

References

1. Network simulator version 3 (ns-3). <https://www.nsnam.org/>
2. Amer, M., Busson, A., Guérin-Lassous, I.: Association Optimization in Wi-Fi Networks based on the Channel Busy Time Estimation. In: IFIP Networking (2018)
3. Cai, K., Blackstock, M., Feeley, M.J., Krasic, C.: Non-intrusive, dynamic interference detection for 802.11 networks. In: Proceedings of the 9th ACM SIGCOMM Internet Measurement Conference, IMC 2009, Chicago, Illinois, USA, November 4-6, 2009, pp. 377–383 (2009)
4. Calhoun, P., Montemurro, M., Stanley, D.: Control and provisioning of wireless access points (capwap) protocol specification. RFC 5415, RFC Editor (2009). URL <http://www.rfc-editor.org/rfc/rfc5415.txt>
5. Chounos, K., Keranidis, S., Korakis, T., Tassiulas, L.: Characterizing the impact of interference through spectral analysis on commercial 802.11 devices. In: IEEE International Conference on Communications, ICC 2017, Paris, France, May 21-25, 2017, pp. 1–6 (2017)
6. Cui, Y., Li, W., Cheng, X.: Partially overlapping channel assignment based on node orthogonality for 802.11 wireless networks. In: 2011 Proceedings IEEE INFOCOM, pp. 361–365 (2011). <https://doi.org/10.1109/INFOCOM.2011.5935182>
7. Dhoutaut, D., Guérin-Lassous, I.: Experiments with 802.11b in ad hoc configurations. In: Proceedings of the IEEE 14th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2003, 7-10 September 2003, Beijing, China, pp. 1618–1622 (2003)
8. Ding, Y., Huang, Y., Zeng, G., Xiao, L.: Channel assignment with partially overlapping channels in wireless mesh networks. In: Proceedings of the 4th Annual International Conference on Wireless Internet, WICON '08, pp. 38:1–38:9. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2008). URL <http://dl.acm.org/citation.cfm?id=1554126.1554173>
9. Grünblatt, R., Guérin-Lassous, I., Simonin, O.: Study of the Intel WiFi Rate Adaptation Algorithm. In: CoRes, 2019, pp. 1–4 (2019)
10. Jang, S., Bahk, S.: A Channel Allocation Algorithm for Reducing the Channel Sensing/Reserving Asymmetry in 802.11ac Networks. *IEEE Trans. Mob. Comput.* **14**(3), 458–472 (2015)
11. Kala, S.M., Reddy, M.P.K., Musham, R., Tamma, B.R.: Interference mitigation in wireless mesh networks through radio co-location aware conflict graphs. *Wireless Networks* **22**(2), 679–702 (2016)
12. Li, W., Zhang, J., Zhao, Y.: Conflict graph embedding for wireless network optimization. In: 2017 IEEE Conference on Computer Communications, INFOCOM 2017, Atlanta, GA, USA, May 1-4, 2017, pp. 1–9 (2017)
13. Magistretti, E., Gurewitz, O., Knightly, E.W.: Inferring and mitigating a link’s hindering transmissions in managed 802.11 wireless networks. In: Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, MOBICOM 2010, Chicago, Illinois, USA, September 20-24, 2010, pp. 305–316 (2010)
14. Paul, U., Kashyap, A., Maheshwari, R., Das, S.R.: Passive Measurement of Interference in WiFi Networks with Application in Misbehavior Detection. *IEEE Trans. Mob. Comput.* **12**(3), 434–446 (2013)
15. Zhou, X., Zhang, Z., Wang, G., Yu, X., Zhao, B.Y., Zheng, H.: Practical Conflict Graphs in the Wild. *IEEE/ACM Trans. Netw.* **23**(3), 824–835 (2015)