

# Towards a fast and efficient strategy to assign channels in WLANs with channel bonding

Amel Chadda

Univ Lyon, ENS de Lyon, Université  
Claude Bernard Lyon 1, Inria, CNRS,  
LIP F-69342, Lyon Cedex 07, France  
amel.chadda@ens-lyon.fr

Marija Stojanova

Univ Lyon, Université Claude Bernard  
Lyon 1, ENS de Lyon, Inria, CNRS,  
LIP F-69342, Lyon Cedex 07, France  
marija.stojanova@ens-lyon.fr

Thomas Begin

Univ Lyon, Université Claude Bernard  
Lyon 1, ENS de Lyon, Inria, CNRS,  
LIP F-69342, Lyon Cedex 07, France  
thomas.begin@univ-lyon1.fr

Anthony Busson

Univ Lyon, Université Claude Bernard  
Lyon 1, ENS de Lyon, Inria, CNRS,  
LIP F-69342, Lyon Cedex 07, France  
anthony.busson@univ-lyon1.fr

Isabelle Guérin Lassous

Univ Lyon, Université Claude Bernard  
Lyon 1, ENS de Lyon, Inria, CNRS,  
LIP F-69342, Lyon Cedex 07, France  
isabelle.guerin-lassous@univ-  
lyon1.fr

## ABSTRACT

With the aim of increasing wireless data rates, IEEE 802.11n introduced the possibility for WLAN nodes to bond two channels into a single channel. However, channel bonding also limits spacial reutilization and complexifies channel assignment. In this paper, we present a fast and efficient solution for channel width selection and channel assignment in 802.11 WLANs using channel bonding. The proposed algorithm uses a novel, graph-centric metric to propose a single channel width for all the APs of the WLAN aiming at avoiding starvation in any of the network's APs. Decoupling the choice of channel width and channel assignment results in a scalable approach that bypasses the usual complexity issues of classic channel assignment schemes. We test the solution's precision in choosing a suited channel width and assignment by comparing its results with those delivered by the ns-3 network simulator. We obtain that, in the large majority of the cases, the choice made by our solution matches the simulation results.

## KEYWORDS

IEEE 802.11, Channel bonding, Channel assignment, Performance, Starvation, Conflict graph, Maximum independent set

### ACM Reference Format:

Amel Chadda, Marija Stojanova, Thomas Begin, Anthony Busson, and Isabelle Guérin Lassous. 2020. Towards a fast and efficient strategy to assign channels in WLANs with channel bonding. In *PE-WASUN '20: ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, November 16–20, 2020, Alicante, Spain*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*PE-WASUN '20, November 16–20, 2020, Alicante, Spain*

© 2020 Association for Computing Machinery.  
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00  
<https://doi.org/10.1145/1122445.1122456>

## 1 INTRODUCTION

With the aim of increasing the data rates of wireless links, IEEE 802.11n introduced the possibility for WLAN (Wireless Local Area Network) nodes to use two channels at the same time to exchange their frames. Bonding two 20 MHz channels into one 40 MHz channel provides a simple way of (slightly more than) doubling the data rate of a wireless link. The practice of channel bonding has been further extended in the following 802.11ac and 802.11ax amendments, with 20 MHz channels being potentially aggregated into 40, 80, or 160 MHz channels.

In addition to increasing the data rate of a wireless link, channel bonding enables (slightly) more efficient use of the (scarce) radio bandwidth by limiting the number of used pilot channels in OFDM. However, bonding two or more channels into one restrains the spatial reutilization of the bandwidth.

To the best of our knowledge, there is currently no simple recipe on how to decide an appropriate width and assignment (a.k.a allocation) for the channels. It is often noted that, in most cases, wider channels favor the aggregate throughput of nodes while narrower channels may represent a better option when fairness among the nodes is a priority (provided the channel assignment was correctly made) (e.g., [1]). However, the issue of selecting the channels and their width remains a complex problem. Clearly, the right choice depends on many factors, and in particular, on the specifics of the WLAN topology.

In this paper, we present the first steps towards a fast and efficient strategy to decide an appropriate channel width and assignment for the APs (Access Points) of a recent 802.11-based WLAN. Unlike an exhaustive approach, our search process is made computationally tractable by two of its features. First, we decouple the original problem into two sub-problems: channel width selection and channel assignment. By doing so, we limit the search to homogeneous assignments, which configure the same channel width for all APs. Second, our proposed solution leverages a novel performance evaluation technique beside classical channel assignment schemes. Overall, our solution helps restrain the number of APs being in severe starvation hence leading to the finding of a good trade-off between aggregate throughput and fairness.

The remainder of the paper is organized as follows. Section 2 provides an overview of the related work. In Section 3, we precisely describe the problem under study. Our proposed solution is presented in Section 4. We evaluate the good behavior of our solution in Section 5 through a number of simulation experiments. Section 6 concludes this paper.

## 2 RELATED WORK

Initially, in the first versions of IEEE 802.11, namely the historical standard and the amendments b/a/g, channels had a fixed width of 20 MHz. In this context, channel assignment (CA) algorithms consisted of assigning non-overlapping channels to APs with the aim of minimizing interference between APs [8, 11, 14, 17].

Since the IEEE 802.11n amendment, APs can aggregate channels (channel bonding). More precisely, the bonding can pertain to two 20 MHz channels in 802.11n, and two, four or eight 20 MHz channels for the most recent amendments (ac/ax) leading to channel widths of 40, 80 and 160 MHz, respectively. While wider channels provide greater physical data rates for transmitting frames, their use increases the number of overlapping channels, interference, and conflicts between APs. Overall, the introduction of channel bonding hardens the CA problem that typically aims at finding an appropriate trade-off between interference and throughput. In particular, the number of possible configurations is significantly increased.

The CA problem with channel bonding is referred to as CB (Channel Bonding) problem in the literature. We categorize solutions to the CB problem through three approaches: *i*) methods that consider a model to evaluate a given assignment and that try to find the one that optimizes an objective function, *ii*) solutions that are based on measurements and adapt their configuration accordingly, and *iii*) algorithms that use machine learning techniques.

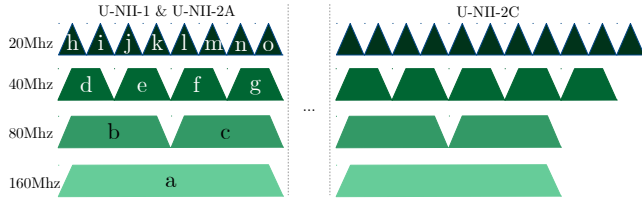
One of the pioneering works presenting a solution for the CB problem belongs to the first approach (model based). The solution named SA (Spectrum Assignment for WLAN) is formulated as an optimization problem for which the authors proposed a distributed resolution [5]. For a given topology, the algorithm aims at minimizing interference between APs while taking into account the preferences of APs for certain channel widths. The authors of [10] present an analytical model that considers both collisions and interference. The CB problem applies to IEEE 802.11ac WLAN and is formulated to optimize the throughput given traffic demand. The optimization problem is then solved through a genetic algorithm. In [6] and [1], the model is based on a continuous time Markov chain (CTMC). The authors of [6] use this model to evaluate the throughput and the system utility for any WLAN topology. It is then combined into an optimization problem for which the authors proposed a heuristic. The authors of [1] use the CTMC on simple topologies to evaluate the performance of the WLAN as a function of the channel widths. Simulations are used for more complex scenarios. They show that an adaptation of the channel bandwidth on a per-packet transmission outperforms single-channel on average but may generate unfair situations where WLANs may starve. A Markov network that models the interaction between the nodes is proposed in [2]. The model and simulations considering IEEE

802.11ac and ax amendments are applied to dense WLANs. Results show that spatial correlations between the nodes significantly impact the performance even between nodes that are not in the sensing range of each other. A renewal process is proposed in [9] to model IEEE 802.11ac and ax WLANs. Results on simple topology allow the authors to derive the performance of channel bonding for these two amendments. Based on these results, a heuristic is proposed to select the channels.

The second category of algorithms uses a data-based approach where channels are set in real-time according to local measurements. In [12], a centralized solution is proposed to solve the CB problem. A controller collects the channel utilization from the APs and updates a matrix whose each element represents the estimated utilization of a given channel when it is assigned to a particular AP. The channel assignment problem is solved by finding a solution that maximizes the sum of the usage of the channel. Another online solution is proposed in [3]. As in [12] the algorithm is based on the activity of the channels. When an AP tests a new channel, it associates a satisfaction score based on what it has been able to send on this channel during a certain period. If the score is satisfactory, the AP remains on this channel, otherwise, it resumes its exploration. Two scenarios are studied in [21] corresponding to an enterprise and a residential WLAN. The CA method considers the constraints imposed by bonding, and attempts to use wider bandwidths as they often result in higher throughput.

Machine learning (ML) techniques offer a promising approach to the CB problem. In [20], a neural network is employed and combined to a Markov chain. This allows the APs to predict the usage of different channels. The assignment is then performed to maintain fairness between channels and APs. Two algorithms based on reinforcement learning are proposed in [15] and [13]. It consists of exploring in real-time new configurations and exploiting the ones that offer good performance. The scenario in [15] is a dense WLAN. The proposed method relies on a graph convolutional network to extract the carrier sensing relationships between APs. A game theory method is then used to collect the training set, on which a neural network is applied to perform the CA. [13] focuses on a multi-armed bandits approach with the Thompson sampling algorithm to select the new configurations to evaluate. In this work, the solution performs both the CA and the associations between APs and stations.

In summary, solutions based either on models or machine learning techniques must explore a tremendous number of assignments and channel widths. They typically improve the current assignment in terms of fairness or throughput, but they typically involve a very limited exploration of the solution space. As for on-line and ML techniques, their exploration phase incurs a cost as new configurations that can be poor in terms of performance have to be tested continuously. In this paper, we split the solution space into its two dimensions (channel and width). By doing so, we significantly reduce the complexity of the CA problem at the cost of having a less fine-grained exploration. Note also that the exploration of solutions is performed using an analytical model so that only the most appropriate found solution is applied to the WLAN. This can prove to be a useful property to avoid disrupting an operational network.



**Figure 1: Available channels in the 5GHz band in Europe [4]. We only label the channels of U-NII-1 and U-NII-2A that we reuse in Section 5.**

### 3 TERMINOLOGY

Before presenting our proposed strategy to select channel width and assignment, we introduce some terminology used to describe WLANs and performance metrics of interest for our study.

#### 3.1 Physical and logical neighbors

We consider an IEEE 802.11 WLAN composed of  $N$  APs that use Static Channel Bonding (SCB) [7]. The APs  $n$  and  $m$  ( $\in \{1, \dots, N\}^2$ ) of the WLAN are said to be *physical neighbors* if they can detect each other's transmissions. Note that the physical neighbors of an AP are mainly determined by the transmission power and sensitivity threshold of its NIC (Network Interface Card) and by the radio wave propagation. We assume that all detection ranges are symmetrical, i.e., if  $n$  is a physical neighbor of  $m$ , then  $m$  is also a physical neighbor of  $n$ . A sample four-AP WLAN is shown in Figure 2a, where AP 3 detects all the other APs, while AP 4 detects only the transmissions of AP 3. Thus AP 3 is the only physical neighbor of AP 4 as depicted in Figure 2b.

Each AP operates either on a basic channel of 20 MHz, or on a bonded channel of 40 MHz, 80 MHz, or 160 MHz. The number of non-overlapping channels for a given width will depend on the chosen width, but also on regional regulations. For example Figure 1 shows the available channels in Europe. In our case, we use the same channel width for all APs, and we use  $w$  to denote its value and  $C_w$  to denote the set of available channels for the width  $w$  MHz. For instance, for  $w = 160$  MHz in Europe and in the U-NII-1 and U-NII-2A bands, we have  $C_w = \{a\}$  and  $|C_w| = 1$ , as shown in Figure 1. It is important to mention that in real-life WLANs, it may happen that two APs that detect each other's transmissions on a 20 MHz channel no longer do so on a wider channel, as the same energy is spread over a wider spectrum.

For the sake of our study, we introduce the notion of *logical neighbors*. Logical AP neighbors are APs that are both physical neighbors and operating on overlapping channels. Clearly, unless there is only one channel in use for the whole WLAN, there are typically much less logical neighbors than physical neighbors thanks to the use of non-overlapping channels.

#### 3.2 Physical and logical conflict graphs

We associate two graphs to any WLAN: its *physical conflict graph*, and its *logical conflict graph*. The former derives only from the pairs of physical neighbors of each AP. Its vertices represent the WLAN's APs and an edge exists between two vertices when the

Notation	Description
$N$	Number of APs in the WLAN
$\mathcal{G}_w(v)$	Logical conflict graph for channel width $w$ and channel assignment described by vector $v$
$C_w$	Set of available channels of width $w$
$L$	Payload length, in bytes
$R$	Physical data rate in Mbps for a given MCS index, number of spatial streams and guard interval length
$a$	Frame aggregation rate, $a \in \{1, \dots, 64\}$
$\gamma_n$	Achievable throughput of AP $n$ in Mbps
$\gamma'_n$	Achieved throughput of AP $n$ in Mbps
$\Gamma$	Aggregate throughput for the WLAN in Mbps
$ST$	Starvation index (number of APs in starvation)
$PF$	Proportional fairness

**Table 1: Principal notations.**

corresponding APs are physical neighbors. Figure 2b depicts the physical conflict graph associated to the WLAN shown in Figure 2a.

The logical conflict graph depends on the chosen channel assignment and hence on the logical neighbors. We use  $\mathcal{G}_w(v)$  to denote the logical conflict graph associated to a channel of width  $w$  and an assignment  $v$  where  $v$  is a vector describing the channels assigned to each AP. For instance,  $v = (b, b, c, b)$  represents a possible channel assignment for our sample four-AP WLAN when the channel assignment is as follows: APs 1, 2, and 4 are all using the same channel of 80 MHz, while AP 3 is using a different channel of 80 MHz too (depicted by the dashed line). Note that, with this channel assignment, AP 3 is no longer in conflict with any of its neighbors and the logical conflict graph involves only one edge between APs 1 and 2. The corresponding conflict graph depicted in Figure 2c is denoted by  $\mathcal{G}_{80}(b, b, c, b)$ .

#### 3.3 Performance metrics

Each AP of the WLAN exchanges traffic with its associated stations. The traffic is characterized by the payload size  $L$  of the frames, the used physical parameters (MCS index, number of spatial streams, guard interval length), and the frame aggregation rate  $a$  (i.e., the average number of MPDUs aggregated in every frame transmission). From these parameters, we can derive the *achievable throughput* of AP  $n$  denoted by  $\gamma_n$  as:

$$\gamma_n = \frac{L \times a}{T_{DCF} + a \times \frac{L + H_{MAC}}{R} + T_{ACK}}, \quad (1)$$

where  $T_{DCF}$  is the sum of the overhead times needed for the DCF procedure [4] including the physical layer header,  $H_{MAC}$  is the MAC header in bytes,  $R$  is the physical data rate resulting from the combination of MCS index, number of spatial streams and guard interval length, and  $T_{ACK}$  is the total time needed to send the acknowledgment frame. The achievable throughput  $\gamma_n$  is simply the throughput AP  $n$  would achieve in downlink if it were the only AP to access its channel. However, this is often not the case and the AP has to share the channel with other logical neighbors APs resulting in a so-called *achieved throughput*. We denote AP  $n$ 's achieved throughput as  $\gamma'_n$ , and it follows that  $\gamma'_n \leq \gamma_n$ . The aggregate throughput of the WLAN, denoted  $\Gamma$ , is simply computed

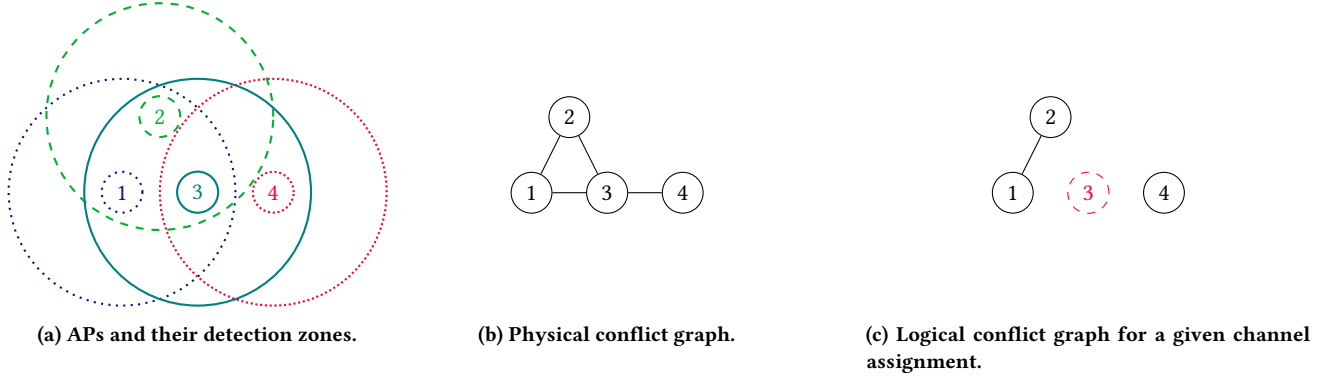


Figure 2: An example of a WLAN with 4 APs.

as the sum of the achieved throughputs of all the APs:

$$\Gamma = \sum_{n=1}^N \gamma'_n. \quad (2)$$

Defining starvation is a bit more problematic as it depends on both the interpretation and the performance goals. In this paper, we consider that an AP is experiencing starvation whenever its ratio  $\frac{\gamma'_n}{\gamma_n} \in [0, 1]$  is lower than some fixed threshold denoted by  $p_{\text{starve}}$ . For example, if  $p_{\text{starve}} = 0.1$ , then an AP is considered in starvation if its achieved throughput is less than 10% of its achievable throughput. We use  $ST$  to denote the number of APs in starvation:

$$ST = \sum_{n=1}^N \mathbb{1}_{\frac{\gamma'_n}{\gamma_n} < p_{\text{starve}}} \quad (3)$$

Furthermore, for the sake of completeness, we also consider the *proportional fairness* metric that we compute here as follows:

$$PF = \sum_{n=1}^N \log \frac{\gamma'_n}{\gamma_n}. \quad (4)$$

Note that  $PF$  has an upper bound  $PF = 0$  for which the solution is deemed most fair (i.e., all the APs have their achieved throughput equal to their achievable throughput) and no lower bound.

Having defined our WLAN representation and the performance metrics of interest for our study, we now detail our proposed solution for radio channel assignment.

## 4 PROPOSED SOLUTION

We now describe our solution for channel selection that circumvents the original problem's complexity by decoupling the channel width and the channel assignment issues.

### 4.1 Maximum Independent set Ratio

Our proposed solution is based on the WLAN's physical conflict graph. It relies on the identification of the APs that are in advantageous positions regarding medium access for transmissions, or equivalently, those most likely to be in starvation because of their location in the conflict graph.

We first recall some graph-theoretic definitions useful for our study. The *maximal independent set* refers to a subset of vertices

that contains no neighbor nodes and cannot be extended by adding more vertices. For example, in our sample four-AP conflict graph in Figure 2b, there are three maximal independent sets:  $\{1, 4\}$ ,  $\{2, 4\}$ , and  $\{3\}$ . A *maximum independent set* denotes the maximal independent set(s) with the highest cardinality. In our example,  $\{1, 4\}$  and  $\{2, 4\}$  are both maximum independent sets. We use  $MMS(\mathcal{G}_w(v))$  to denote the set containing all the maximum independent sets of  $\mathcal{G}_w(v)$ .

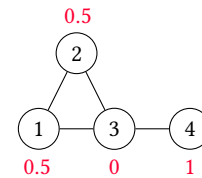
We can now introduce a new quantity that we specifically devise for our study: the *Maximum Independent set Ratio (MIR)*.  $MIR$  is calculated for each of the graph's vertices as the proportion of maximum independent sets to which the vertex belongs. Therefore, for vertex  $n$ , we have:

$$MIR(n, \mathcal{G}_w(v)) = \frac{|K|}{|MMS(\mathcal{G}_w(v))|}. \quad (5)$$

where  $K$  is the subset of  $MMS(\mathcal{G}_w(v))$  that verifies  $n \in K$ .

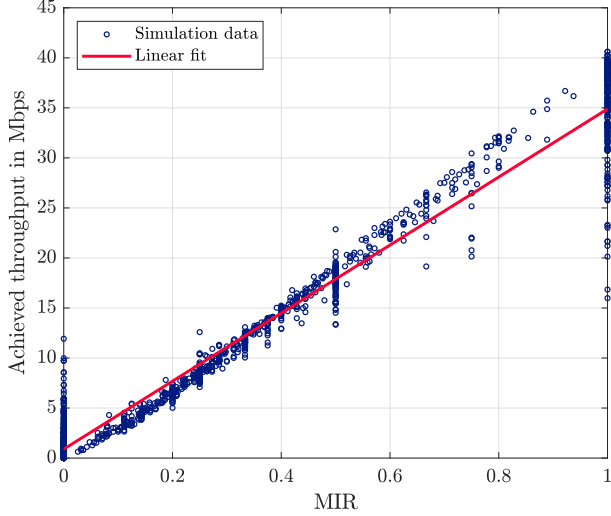
The maximum independent sets and the  $MIR$  values of any graph and any vertex can be obtained using the Bron-Kerbosch algorithm [19]. We discuss the complexity of the latter algorithm in Section 4.3.

As an example, the  $MIR$  values for the vertices representing our four-AP sample network when using a single channel of 160 MHz for all APs (i.e.,  $v = (a, a, a, a)$ ) are shown in Figure 3. AP 4 belongs to all maximum independent sets and so it has an  $MIR(4, \mathcal{G}_{160}(a, a, a, a)) = 1$ . Conversely, APs 1 and 2 belong to half of the maximum independent sets, so we have  $MIR(1, \mathcal{G}_{160}(a, a, a, a)) = MIR(2, \mathcal{G}_{160}(a, a, a, a)) = 0.5$ . Finally, we have  $MIR(3, \mathcal{G}_{160}(a, a, a, a)) = 0$  since AP 3 does not belong to any maximum independent set.

Figure 3: Calculated  $MIR$ s for a given logical conflict graph.

Interestingly, the  $MIR$  of vertex  $n$  is a major factor in determining its achieved throughput  $\gamma'_n$ . Let us consider a set of 60

randomly generated physical conflict graphs of WLANs containing between 8 and 30 APs and with average AP degrees ranging from 2.5 to 8.3. Assuming APs in saturation with only downlink traffic to their single associated station using MCS index 5, a same channel of width  $w = 20$  MHz, a frame aggregation rate  $a = 4$ , and a payload  $L = 1500$  bytes, we use the ns-3 discrete-event network simulator [16] to evaluate the achieved throughput  $\gamma'_n$  of each AP. Figure 4 depicts the AP's achieved throughput as a function of the *MIR* of every AP of these graphs. We observe that there is a strong correlation between an AP's *MIR* and its achieved throughput  $\gamma'_n$ .



**Figure 4: The achieved throughput as a function of *MIR* obtained by simulating the behavior of hundreds of APs for 60 different WLANs.**

This observation is the cornerstone of our proposed solution. Indeed, for a given logical conflict graph, we can easily calculate the *MIR* value of each AP that, in turn, helps us estimate its attained throughput and hence predict whether that AP is in starvation or not. To do that, we naturally resort to a linear regression as follows:

$$\gamma' = \beta_0 + \beta_1 \times \text{MIR}. \quad (6)$$

where  $\beta_0$  and  $\beta_1$  are found to be equal to 0.89 and 34.00, respectively.

## 4.2 Channel width and assignment algorithm

In our solution, we perform the search for the channel assignment in the worst-case scenario, namely when all APs are saturated (permanently needing to access the channel). We also rely on the assumption, as it was shown by in [1], that wider channels favor the aggregate throughput of the WLAN, while smaller and separate channels avoid starvation. Hence, our solution iterates on the possible channel widths, starting with the widest one, i.e., 160 MHz. For each channel width, we use the Tabu Search algorithm [18] to compute a  $k$ -coloring of the physical conflict graph, where  $k$  is the number of available channels for the considered channel width. In a nutshell, the Tabu search attempts to provide a channel assignment that minimizes the number of edges in the resulting logical conflict graph. We then calculate the *MIR* of each AP and estimate

its attained throughput using the linear regression model. At this stage, we are able to compute *ST* that evaluates the number of APs in starvation. If this latter is larger than one, we divide the channel width in half and multiply the number of available channels by two. With this new setting, we run again the same procedure: coloring the graph, calculating the APs' *MIR*s and *ST* metrics, etc. The procedure ends when a channel width for which there are no starving APs is found, or when the minimum channel width of 20 MHz is reached. The pseudo-algorithm of the complete solution is given in Algorithm 1.

---

### Algorithm 1 Solution to select channel width and assignment

---

- 1: **Input:** physical conflict graph, number of vertices  $N$ , starvation threshold  $p_{\text{starve}}$
- 2: **Output:** selected channel width  $w$ , channel assignment  $v$

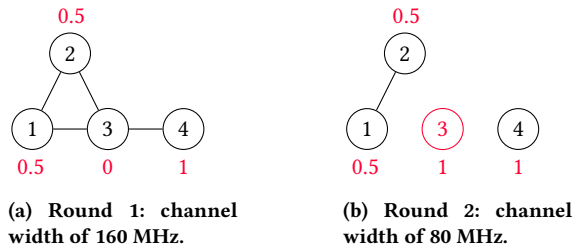
---

- 3:  $w \leftarrow 160$
- 4:  $k \leftarrow \lfloor C_{160} \rfloor$  // # of channels for the current width
- 5: **while**  $w \geq 20$  **do**
- 6:    $v \leftarrow$  compute a  $k$ -coloring channel assignment (Tabu)
- 7:   compute  $\text{MIR}(n, \mathcal{G}_w(v)), \forall n \in \{1, \dots, N\}$  using Eq. 5
- 8:   compute  $\gamma'_n, \forall n \in \{1, \dots, N\}$  using Eq. 6
- 9:   compute *ST* using Eq. 3
- 10:   **if**  $ST > 0$  **then**
- 11:      $w \leftarrow w / 2$
- 12:      $k \leftarrow \lfloor C_w \rfloor$
- 13:   **else**
- 14:     **return**  $(w, v)$
- 15:   **end if**
- 16: **end while**
- 17: **return**  $(w, v)$

---

Note that Algorithm 1 has a single tuning parameter: the starvation threshold,  $p_{\text{starve}}$ . If its value is too large, this could cause the algorithm to erroneously suppose APs in starvation and to wrongfully reject an otherwise appropriate channel assignment. Conversely, a too small value for  $p_{\text{starve}}$  can lead the algorithm to overlook APs in starvation and lead to channel assignment causing starvation.

In the interest of clarity, Figure 5 shows the results returned by Algorithm 1 on the sample four-AP WLAN using a starvation threshold  $p_{\text{starve}} = 0.125$ . For the sake of this example, we assume that there is only a single 160 MHz channel so that  $\lfloor C_{160} \rfloor = 1$ . The main round of the algorithm was executed twice, namely for a channel width of 160 MHz and then 80 MHz. In the first round, we have  $\text{MIR}(3, \mathcal{G}_{160}(a, a, a, a)) = 0$  so that the attained throughput of AP 3  $\gamma'_3$  is estimated to be 0. Thus, we have  $ST > 0$  and the algorithm rejects the channel width of 160 MHz. The algorithm now considers a channel width of 80 MHz. For this second round, we obtain  $\text{MIR}(3, \mathcal{G}_{80}(b, b, c, b)) = 1$  leading to an attained throughput of 20 Mbps, meaning that, at least according to our Algorithm, there should be no APs in starvation as the APs 1, 2, and 4 have *MIR*s of 0.5, 0.5 and 1, respectively, which are all above the starvation threshold. Therefore, the algorithm ends and returns  $w = 80$  MHz and  $v$  for the channel assignment as its solutions.



**Figure 5: Logical conflict graph and  $MIR$  values computed by Algorithm 1 on the example of the four-AP WLAN of Figure 2b.**

### 4.3 Complexity analysis

The complexity of the proposed solution is highly dependent on the complexity of the underlying algorithms. The Tabu Search algorithm that we use to find a  $k$ -coloring has an  $O(D_{max}N^4)$  complexity (when the objective function is to minimize the number of edges in the logical conflict graph), where  $D_{max}$  is the maximum degree in the graph [18]. Bron-Kerbosch’s algorithm that is used to find the  $MIR$  ratios has an  $O(3^{\frac{N}{3}})$  complexity [19]. All other parts of our algorithm have a lower complexity, meaning that the overall complexity of the proposed solution is  $O(3^N)$ .

It should be noted that the use of the Tabu Search and the Bron-Kerbosch algorithms is fully independent of the rest of the solution. Moreover, other objective functions than the minimization of the number of edges in the logical conflict graph can be used as well. These possible modifications, as well as the choice of starvation threshold, create an easily adaptable framework suited for different WLAN performance goals.

## 5 NUMERICAL RESULTS

In this section, we apply our proposed solution to a large set of randomly generated physical conflict graphs and study its behavior for different values of  $p_{starve}$ . We then evaluate the solution’s accuracy by comparing its results with those provided by the ns-3 simulator. Note that for the sake of simplicity and without loss of generality, we assume in this section that the pairs of physical neighbors in a conflict graph do not depend on the width of their channel. The set of available channels contains all the channels from the U-NII-1 and U-NII-2A bands available in Europe (see Figure 1), i.e., one channel of 160 MHz, two of 80 MHz, four of 40 MHz, and eight channels of 20 MHz.

### 5.1 Graph generation and simulation setup

We consider a set of 105 physical conflict graphs as our validation dataset. The graphs are randomly generated with average degrees from 2.5 to 8.3 and consist of 8 to 30 APs. We run our proposed solution on every graph of this set and obtain in return a channel assignment vector  $v$  associated to a given channel width  $w$ . We repeat the same experiment for three different values of  $p_{starve}$ : 0.125, 0.25, and 0.5.

On the other hand, for every possible channel width, we input the channel assignment found by our algorithm into the ns-3 simulator and obtain the APs’ achieved throughputs. Each simulation

is run for 60 seconds of simulated time in which only the APs are generating downlink traffic to their single associated station. The APs aggregate four MPDUs in every frame transmission using MCS index 5 and a payload of 1500 bytes.

### 5.2 Influence of $p_{starve}$ on the chosen channel width

We begin by examining the frequency at which a channel width of 160, 80, 40, and 20 MHz was returned by our proposed solution across the 105 examples of our dataset. Figure 6 shows the corresponding histograms for  $p_{starve}$  values of 0.125, 0.25, and 0.5. We first observe that, among the 105 examples, our algorithm never selects a channel width of 160 MHz. This means that our algorithm supposes that on a 160 MHz channel there will always be at least one AP in starvation. Second, we observe that the mode (most frequent value) of the histograms is the channel width of 40 MHz, represented in more than 70% of the cases. Widths of 80 MHz and 20 MHz are returned at much lower frequencies. Finally, Figure 6 shows how the channel width selected by our algorithm varies with the starvation threshold,  $p_{starve}$ . As expected, lower values of  $p_{starve}$  will tend to result in selecting wider channels, and vice versa.

### 5.3 Accuracy evaluation

Having verified that our algorithm responds differently to different physical conflict graphs, we now evaluate the precision of its decisions. To check if the network simulator agrees with our solution, we simulate the best logical conflict graph found by Tabu search for each channel width and compute the associated number of APs in starvation, namely  $ST$ , estimated by the simulator.

To ease the reading of the corresponding results, we represent the results as a number of hits and misses. A hit is obtained whenever our solution and the simulator agree on the selected channel width. Otherwise, disagreement leads to a miss. Tables 2a, 2b, and 2c summarize the number of hits and misses for  $p_{starve}$  values of 0.125, 0.25, and 0.5, respectively. Table 2a shows that for  $p_{starve} = 0.125$  there is only one conflict graph for which our solution recommendation and that of the simulator differ (80 MHz vs 40 MHz). For  $p_{starve} = 0.25$ , the responses of our solution and of the simulator are in full agreement as shown by Table 2b. On the other hand, when considering a threshold for starvation  $p_{starve}$  equal to 0.5, Table 2c indicates that our proposed solution leads to many misses. Indeed, with this setting, our solution recommends 90 (18 + 72) times a channel width of 40 MHz but out of these 90, only 18 are also recommended by the simulator. The high number of misses can be explained by the uncertainty in predicting AP  $n$ ’s throughput when  $MIR(n, \mathcal{G}_w(v)) = 0.5$ , as shown in Figure 3. However, it is questionable whether considering such a high value for the starvation threshold is a reasonable choice, as it requires that all APs occupy the medium 50% of the time. Overall, our results can be summarized as follows. First, we recommend using a starvation threshold  $p_{starve}$  in the range of 0.125 and 0.25. Second, given a relevant value of  $p_{starve}$ , our algorithm provides virtually the same channel selection as the simulator while being only based on the physical conflict graph.

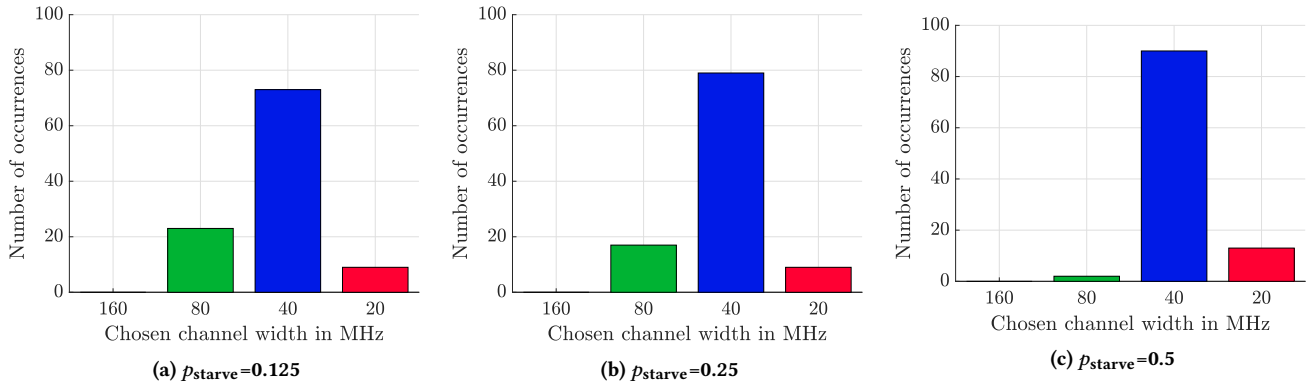


Figure 6: Histograms of the channel width chosen by our proposed solution for the 105 considered physical conflict graphs.

our solution \ simulation	80 MHz	40 MHz	20 MHz
80 MHz	22	1	0
40 MHz	0	73	0
20 MHz	0	0	9

(a)  $p_{\text{starve}} = 0.125$

our solution \ simulation	80 MHz	40 MHz	20 MHz
80 MHz	17	0	0
40 MHz	0	79	0
20 MHz	0	0	9

(b)  $p_{\text{starve}} = 0.25$

our solution \ simulation	80 MHz	40 MHz	20 MHz
80 MHz	0	2	0
40 MHz	0	18	72
20 MHz	0	0	13

(c)  $p_{\text{starve}} = 0.5$

Table 2: Distribution of hits and misses of our solution: the numbers on the main diagonal represent the cases when our solution’s channel width agrees with the simulation results. In the other cases, the simulator provided a different most favorable channel width.

An interesting question is how well the channel selection provided by our algorithm ranks in terms of proportional fairness, as defined in Eq. 4. To address this point, we rely on the simulator to discover the channel selection that maximizes the proportional fairness among the channel allocation provided by our algorithm for all channel widths. The obtained results show that for 103 out of the 105 graphs, choosing a 40 MHz channel offers the best proportional fairness. On the other hand, as shown by Figure 6, in most cases, our approach also returned a channel width of 40 MHz (the actual value depends on the  $p_{\text{starve}}$  setting but is never lower than 70%). This may represent an interesting avenue of research for our approach that would be the focus of further investigation.

## 6 CONCLUSIONS

In this paper, we presented a fast and efficient solution for channel width selection and channel assignment in 802.11 WLANs using channel bonding. The proposed algorithm chooses a single channel width for all the APs of the WLAN that aims to avoid starvation in any of the network’s APs.

By introducing a novel, graph-centric metric, and by decoupling the channel width selection and the channel assignment, we offer a scalable approach that bypasses the usual complexity issues of classic channel assignment schemes. The approach is also fully adaptable to different performance goals, as it is a collection of several algorithms that can be modified separately and independently.

We test the solution’s ability in choosing a suited channel width and assignment by comparing its results with those delivered by the ns-3 network simulator. We obtain that in the large majority of the cases, the choice made by our solution matches the simulation results. Moreover, the selected channel width seems to not only minimize starvation but also maximize proportional fairness.

As future works, we intend to improve the numerical validation of our solution robustness by considering more realistic traffic. We also plan to further evaluate the ability of our solution to maximize other performance metrics such as proportional fairness.

## REFERENCES

- [1] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. 2019. To overlap or not to overlap: Enabling channel bonding in high-density WLANs. *Computer Networks* 152 (2019), 40 – 53. <https://doi.org/10.1016/j.comnet.2019.01.018>
- [2] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. 2020. Dynamic Channel Bonding in Spatially Distributed High-Density WLANs. *IEEE Transactions on Mobile Computing* 19, 4 (2020), 821–835.
- [3] S. Barrachina-Muñoz, F. Wilhelmi, and B. Bellalta. 2020. Online Primary Channel Selection for Dynamic Channel Bonding in High-Density WLANs. *IEEE Wireless Communications Letters* 9, 2 (2020), 258–262.
- [4] M. S. Gast. 2013. *802.11 ac: a survival guide: Wi-Fi at gigabit and beyond*. O’Reilly Media, Inc.
- [5] J. Herzen, R. Merz, and P. Thiran. 2013. Distributed spectrum assignment for home WLANs. In *2013 Proceedings IEEE INFOCOM*. 1573–1581.
- [6] C. Kai, Y. Liang, X. Hu, Z. Liu, and L. Wang. 2019. An effective channel allocation algorithm to maximize system utility in heterogeneous DCB WLANs. *Computer Networks* 153 (2019), 23 – 35. <https://doi.org/10.1016/j.comnet.2019.01.043>
- [7] Caihong Kai, Yuting Liang, Tianyu Huang, and Xu Chen. 2017. To bond or not to bond: An optimal channel allocation algorithm for flexible dynamic channel bonding in WLANs. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE.
- [8] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot. 2007. Measurement-Based Self Organization of Interfering 802.11 Wireless

- Access Networks. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*.
- [9] S. Khairy, M. Han, L. X. Cai, Y. Cheng, and Z. Han. 2019. A Renewal Theory Based Analytical Model for Multi-Channel Random Access in IEEE 802.11ac/ax. *IEEE Transactions on Mobile Computing* 18, 5 (2019), 1000–1013.
- [10] S. Lee, T. Kim, S. Lee, K. Kim, Y. H. Kim, and N. Golmie. 2019. Dynamic Channel Bonding Algorithm for Densely Deployed 802.11ac Networks. *IEEE Transactions on Communications* 67, 12 (2019), 8517–8531.
- [11] D.J. Leith, P. Clifford, V. Badarla, and D. Malone. 2012. WLAN channel selection without communication. *Computer Networks* (2012). <https://doi.org/10.1016/j.comnet.2011.12.015>
- [12] T. H. Lim, W. S. Jeon, and D. G. Jeong. 2016. Centralized channel allocation scheme in densely deployed 802.11 wireless LANs. In *2016 18th International Conference on Advanced Communication Technology (ICACT)*. 249–253.
- [13] A. López-Raventós and B. Bellalta. 2020. Concurrent decentralized channel allocation and access point selection using multi-armed bandits in multi BSS WLANs. *Computer Networks* 180 (2020), 107381. <https://doi.org/10.1016/j.comnet.2020.107381>
- [14] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. 2006. A Client-Driven Approach for Channel Management in Wireless LANs. In *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*.
- [15] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura. 2019. Deep Reinforcement Learning-Based Channel Allocation for Wireless LANs with Graph Convolutional Networks. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. 1–5.
- [16] ns3 [n.d.]. The Network Simulator ns-3. <https://www.nsnam.org/>.
- [17] E. Rozner, Y. Mehta, A. Akella, and L. Qiu. [n.d.]. Traffic-Aware Channel Assignment in Enterprise Wireless LANs. In *2007 IEEE International Conference on Network Protocols*.
- [18] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao. 2008. Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks. *IEEE Trans. Mob. Comput.* (2008). <https://doi.org/10.1109/TMC.2008.70>
- [19] E. Tomita, A. Tanaka, and H. Takahashi. 2004. The Worst-Case Time Complexity for Generating All Maximal Cliques. In *Computing and Combinatorics, 10th Annual International Conference*. Springer. [https://doi.org/10.1007/978-3-540-27798-9\\_19](https://doi.org/10.1007/978-3-540-27798-9_19)
- [20] J. Webber, A. Mehbodniya, K. Yano, and Y. Suzuki. 2019. Optimized WLAN Channel Allocation based on Gibbs Sampling with Busy Prediction using a Probabilistic Neural Network. In *2019 International Conference on Communications, Signal Processing, and their Applications (ICCSIPA)*. 1–5.
- [21] A. Zakrzewska and L. Ho. 2019. Dynamic Channel Bandwidth Use through Efficient Channel Assignment in IEEE 802.11ac Networks. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*.