SAW: Spectrum Assignment for WLANs

Julien Herzen
EPFL, Switzerland
julien.herzen@epfl.ch

Ruben Merz
Swisscom, Switzerland
ruben.merz@swisscom.com

Patrick Thiran
EPFL, Switzerland
patrick.thiran@epfl.ch

ABSTRACT

[1] We consider the problem of jointly allocating channel center-frequencies and bandwidths for IEEE 802.11 wireless LANs (WLANs). The bandwidth used on a link significantly affects both the capacity experienced on this link and the interference produced on neighboring links. Therefore, when jointly assigning both center frequencies and channel widths, a trade-off must be found between interference mitigation and the potential capacity offered on each link. We study this trade-off and present SAW (spectrum assignment for WLANs), a decentralized algorithm that finds efficient configurations.

SAW is tailored for 802.11 home networks. It is distributed, online and transparent. It does not require a central coordinator and it constantly adapts the spectrum usage without disrupting network traffic. The algorithm is decentralized and self-organizing; it provably converges towards efficient spectrum allocations. We evaluate SAW using both simulation and a deployment on an indoor testbed composed of off-the-shelf 802.11 hardware. We observe that it dramatically increases the overall network efficiency and fairness, even when some Access Points (APs) do not behave socially.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

General Terms

Algorithms, Design, Measurement, Performance

1. INTRODUCTION

Current WLANs offer the possibility of adapting both their operating channel center-frequency and bandwidth. The channel center-frequency determines where the nodes operate in the available spectrum, and the channel width determines how much spectrum they occupy. Although the problem of channel assignment alone has been extensively studied in the literature (see e.g., [4,5]), the problem of assigning the two parameters jointly has received limited treatment (see e.g., [7] for an example of a centralized algorithm). Yet, considering the channel width adds an important degree of freedom [1] and drastically changes the nature of the problem. Usually, for links with a large enough SNR, using a wider bandwidth benefits the achievable throughput [1]. There is thus a trade-off between the amount of spectrum offered on each link, and the likelihood that two neighboring links interfere if they use wide – potentially overlapping – spectral bands.

In Section 2, we present a distributed algorithm that converges to optimal solutions, in the sense of an explicit formulation of this interference versus capacity trade-off. In Section 3, we present some results obtained both from simulation and testbed experiments. Finally, we conclude in Section 4.

2. SAW ALGORITHM

2.1 Model

Denote by l and k two WiFi links. We compute $I_l(k)$, the (asymmetric) interference produced by k on l as $I_l(k) := \mu_k \cdot IF(k,l)$, where μ_k is the average proportion of time during which k transmits, and $IF(k,l)$ is the interference factor (frequency overlap) defined in [6]. Similarly, if A and B denote two basic service sets (BSSs)[2], the interference produced by B on A is defined as

$$I_A(B) := \sum_{l \in A} \sum_{k \in B} I_l(k).$$

We write f_A and b_A for the center-frequency and the bandwidth used by BSS A, respectively. Finally, we write $\mathcal{N}_A$ for the set of neighbor BSSs of A.

2.2 Algorithm

We formulate the interference versus capacity trade-off explicitly, and seek to solve the following optimization problem:

$$\mathsf{minimize} \quad \mathcal{E} := \sum_{A} \sum_{B \in \mathcal{N}_A} I_A(B) + c \cdot \sum_{A} \mathrm{cost}_A(b_A), \qquad (1)$$

where the minimization is over all the possible combinations of center frequencies and bandwidths. The first term accounts for the total interference level present in the network, whereas the second term encourages the usage of bandwidths that offer more capacity. The constant c is a weighting parameter between the two terms. The function $\mathrm{cost}_A(b_A)$ denotes the "cost" attributed by the AP of BSS A to the usage of bandwidth b_A. For links having large enough

---

[1] A full-length version of this work has been published in [3]. The present document contains additional simulation results, presented in Section 3.1.3, which treat situations where some access points behave selfishly.

[2] A BSS is a set of links that contain one AP.

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

Algorithm 1: SAW algorithm at the AP of BSS A

```
1 **Initialization:**
2 Pick a random configuration $(f_A, b_A)$
3 **After random, exponentially distributed time intervals:**
4 Pick a random configuration $(f_{\text{new}}, b_{\text{new}})$
5 Measure $e_1 := \sum_{B \in \mathcal{N}_A} (I_A(B) + I_B(A)) + \text{cost}_A(b_A)$ if $A$ uses $(f_A, b_A)$
6 Measure $e_2 := \sum_{B \in \mathcal{N}_A} (I_A(B) + I_B(A)) + \text{cost}_A(b_{\text{new}})$ if $A$ uses $(f_{\text{new}}, b_{\text{new}})$
7 Compute
      $$\beta_T = \begin{cases} 1 & \text{if } e_2 < e_1 \\ \exp\left(\frac{e_1 - e_2}{T}\right) & \text{otherwise} \end{cases}$$
8 Set $(f_A, b_A) = (f_{\text{new}}, b_{\text{new}})$ with probability $\beta_T$
```

SNRs, using a larger bandwidth offers better throughput [1]. For this reason, in the remainder, we use $\text{cost}_A(b_A) = 1/b_A$ to encourage wider bandwidths. Note however that if a BSS A has some links with a poor SNR it may choose to use a different function $\text{cost}_A(b_A)$.

SAW is presented in Algorithm 1, as it runs at the AP of a BSS A. It is a distributed Metropolis sampler for the channel center-frequency and bandwidth. If all APs run SAW, the allocation of channels and bandwidths across all APs get asymptotically arbitrarily close (for T small enough) to the global optimum of Problem (1).

3. PERFORMANCE EVALUATION

3.1 Simulation Results

Before evaluating SAW on an indoor 802.11 testbed in Section 3.2, by using simulation, we investigate its self-organization properties on large ecosystems of interfering WLANs. We consider either random distributions of N BSSs (composed of one AP and two clients) on the unit square, or within N cells of a regular grid (see [3] for more information on the simulation settings). Unless otherwise specified, we use $c = 1$ and $T = 0.1$, with $N = 100$ BSSs, and we show the median values obtained over 50 independent executions. We consider a scenario identical to the 2.4 GHz ISM band, with 11 possible center frequencies in 70 MHz of total spectrum, and four different channel bandwidths (5, 10, 20 and 40 MHz). The traffic is downlink, from the AP to their clients. At initialization, each BSS picks a random channel and uses the largest width.

3.1.1 *Interference vs. Capacity*

In general, it is expected that the ideal weight c between the two terms of Equation (1) should depend on the network spatial density: For sparse networks, the links have few neighbors and they can freely use the spectrum, whereas for dense networks, more weight should be given to interference mitigation. We evaluate the influence of c in Figure 1, as a function of the network spatial density (in this case, all APs use the cost function $\text{cost}_A(b_A) = c/b_A$, for different values of c). As expected, $c = 0$ yields the best performance for dense networks, and a large value of c yields the best performance for sparse networks. However, a low but non-zero value of c performs best *in all regimes*. This means that a unique weight can be used by all APs to balance interference mitigation with spectrum usage, irrespective of the network density.

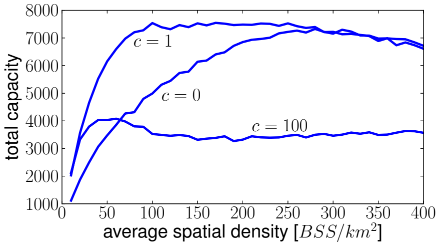

Figure 1: Sum of link capacities (after 30 iterations per AP on average) as a function of network spatial density, for several values of the weighting coefficient c.

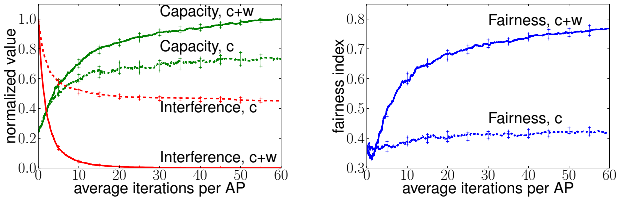

Figure 2: Capacity, interference and fairness as functions of the number of iterations. We show the values obtained when SAW tunes both the channel center-frequency and bandwidth (c+w), and when it tunes only the center-frequency (c). The values for the capacity and interference are normalized with respect to their observed maximum for convenience.

3.1.2 *Gains of Joint Allocation*

In Figure 2, we show the sum of link capacities, the interference (the first term of Equation (1)) and the Jain fairness index (across BSSs) as functions of the number of iterations of the algorithm. The results are displayed either when SAW tunes only the channel center-frequency of the APs, or when it tunes the center-frequency jointly with the bandwidth. All three metrics are drastically improved after only a couple of tens of iterations per AP. In addition, tuning the two parameters jointly provides significant improvements compared to the case where only the center-frequency is adapted.

3.1.3 *Selfish APs*

In lines 5 and 6 of Algorithm 1, when deciding on whether to adopt a new configuration or not, an AP considers not only the interference received from its neighbors ($I_A(B)$), but also the interference that itself creates on its neighbors ($I_B(A)$). Note that in general, due to the asymmetric nature of wireless interference, these two interference terms need not be equal. With this way of sampling configurations, the APs take into account the interference that they produce on their neighbors, wich might go against their immediate best interest. This "equity" is necessary for establishing the convergence of the algorithm (see [3]), but it could be harmed if some APs do not behave socially (i.e., according to the legacy algorithm). We now consider such a scenario, where some subset(s) of APs run a selfish version of the algorithm. These selfish APs do not take the interference created on their neighbors into account and, for these APs, the first term of lines 5 and 6 is replaced by $\sum_{B \in \mathcal{N}_A} I_A(B)$.

Figure 3 (left) shows the average capacity and interference experienced by the two categories of BSSs – with selfish and legacy APs – for varying proportion of APs running the selfish version of the algorithm. As expected, the maximum capacity is obtained when all nodes behave according to the legacy algorithm. For both

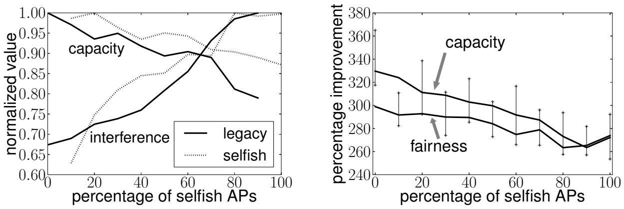

Figure 3: Left: achieved capacity and interference (per AP), for both selfish and legacy APs. Right: improvement (compared to random allocations of fixed-width channels) of capacity and fairness as functions of the proportion of selfish APs in the network.

selfish and legacy APs, the capacity decreases with the proportion of selfish APs, due to increased interference levels. In general, the selfish APs obtain a slightly better capacity than the legacy APs. However, the performance loss due to the presence of selfish APs is not drastic and, even when 100% of the APs are selfish, they still achieve 87% of the capacity achieved when all APs are legacy. This is confirmed in Figure 3 (right), where we plot the percentage of improvement of capacity and fairness obtained with our algorithm (compared to random allocations of fixed-width channels), with a varying proportion of selfish APs. Although the presence of selfish APs slightly decreases performance, the improvements still remain above $2.6\times$ for both capacity and fairness.

We explain these large gains in the presence of selfish APs by observing that although the interference levels experienced by two neighboring links l and k are asymmetric, they are often strongly correlated. Therefore, if (the AP of) link k seeks to selfishly minimize its own interference level, it is likely to also decrease the interference that it produces on l. As a consequence, the price of anarchy remains limited in these cases, and the algorithm shows some robustness to the presence of selfish APs. A similar conclusion has been reached in [2] in the context of Gaussian interference games.

3.2 Testbed Results

We implemented SAW on a testbed of 21 wireless nodes, which form 10 BSSs spread over an entire floor of the EPFL BC building. Each node is composed of off-the-shelf hardware, running OpenWrt with the ath9k driver. We perform the experiments on the 2.4 GHz band using 802.11g with three different bandwidths (5, 10 and 20 MHz). SAW is implemented in C++ as four new elements of the *Click* modular router. More information on the implementation are provided in [3]. We use $c = 1$, $T = 0.1$, and a mean wake-up time of 4 minutes for the algorithm.

3.2.1 Performance

We perform several experiments with UDP and TCP traffic. Traffic is backlogged and downlink, from the APs to their clients. This represents a frequent use case where all the capacity is used, for instance when several clients are downloading simultaneously from the Internet. All BSSs start in channel 6 with a bandwidth of 20MHz. As a benchmark, we use a centralized channel allocation algorithm (based on graph coloring) for attributing three non-overlapping (fixed-width) channels (1, 6 and 11) to each BSS. This procedure is centralized and is a reasonable upper-bound of what can possibly be achieved with an unplanned deployment.

Figure 4 shows the average sum and the standard deviations (over 20 independent runs) of the throughput achieved by each link. We also show the average obtained with the benchmark. In each scenario, SAW starts at 600 seconds. In general, SAW settles to spectrum assignments that are equivalent or better than centralized graph-coloring. The extra gain is due to the joint adaptation of both the center-frequency and the bandwidth operated by SAW. In these experiments, much of the gain already comes after one or two iterations of SAW per BSS, and the algorithm finds efficient allocations after approximately 3 iterations per BSS on average.

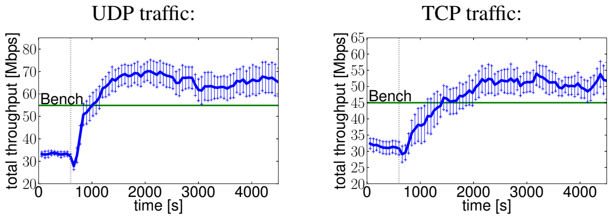

Figure 4: Sum of the link throughput's obtained by the 10 BSSs with downlink traffic. SAW is started at 600 seconds. The "Bench" line is the average throughput obtained with a centralized graph coloring approach that uses the 3 non-overlapping channels with a width of 20 MHz.

4. CONCLUSION

We have presented SAW, a decentralized algorithm that finds efficient variable-width spectrum configurations for WLANs. The spectrum allocation problem is formulated as the global minimization of a weighted sum, composed of neighbor interactions and local bandwidth preferences. We have observed that a single weight value enables the algorithm to solve the capacity-interference tradeoff, irrespective of the network spatial density. We have conducted performance evaluation using simulations and testbed experiments. SAW provides drastic capacity and fairness improvements, even when a large proportion of the APs behave selfishly. Thanks to its underlying Metropolis formulation, where only one new configuration is sampled at a time, SAW scales nicely with the total number of available channels and bandwidths. This property suggests that some of the concepts presented in this paper could be applicable to white-space networks.

5. REFERENCES

[1] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl. A case for adapting channel width in wireless networks. In *ACM SIGCOMM 08*, pages 135–146, 2008.

[2] R. Etkin, A. Parekh, and D. Tse. Spectrum sharing for unlicensed bands. *Selected Areas in Communications, IEEE Journal on*, 25(3):517–528, 2007.

[3] J. Herzen, R. Merz, and P. Thiran. Distributed spectrum assignment for home wlans. In *IEEE Infocom*, 2013.

[4] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot. Measurement-based self organization of interfering 802.11 wireless access networks. In *INFOCOM 07*, pages 1451–1459, 2007.

[5] D. J. Leith, P. Clifford, V. Badarla, and D. Malone. WLAN channel selection without communication. *Computer Networks*, Jan. 2012.

[6] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh. Partially overlapped channels not considered harmful. In *SIGMETRICS 06*, 2006.

[7] S. B. Shravan Rayanchu, Vivek Shrivastava and R. Chandra. FLUID: Improving throughputs in enterprise wireless lans through flexible channelization. In *ACM Mobicom 11*, 2011.