Improving the Spatial Reuse in IEEE 802.11ax WLANs: A Multi-Armed Bandit Approach

Anthony Bardou
anthony.bardou@ens-lyon.fr
Univ Lyon, ENS de Lyon, Université Claude Bernard Lyon 1, Inria, CNRS, LIP
Lyon, Rhône-Alpes, France

Thomas Begin
thomas.begin@ens-lyon.fr
Univ Lyon, ENS de Lyon, Université Claude Bernard Lyon 1, Inria, CNRS, LIP
Lyon, Rhône-Alpes, France

Anthony Busson
anthony.busson@ens-lyon.fr
Univ Lyon, ENS de Lyon, Université Claude Bernard Lyon 1, Inria, CNRS, LIP
Lyon, Rhône-Alpes, France

ABSTRACT

The latest amendment 802.11ax to the IEEE 802.11 standard, better known by its commercial name Wi-Fi 6, includes a feature that aims at improving the spatial reuse of a channel: each device can adapt its Clear Channel Assessment sensitivity threshold and its transmission power. In this paper, we use the Multi-Armed Bandit (MAB) framework to propose a centralized solution to dynamically adapt these parameters. We propose a new approach based on a Gaussian mixture to sample new network configurations, a specific reward function that prevents starvations when maximized, as well as a method based on Thompson Sampling to select the best network configuration. We evaluate our solution using the network simulator ns-3 and different topologies. Simulation results confirm the large benefits that 802.11ax may bring to spatial reuse. They also demonstrate the efficiency of our solution in finding appropriate parameter configurations that significantly improve the quality of service of the networks.

CCS CONCEPTS

• **General and reference** → **Performance**; • **Networks** → **Network control algorithms**; **Network simulations**; **Wireless local area networks**; • **Theory of computation** → **Bayesian analysis**; **Reinforcement learning**.

KEYWORDS

Machine Learning; Thompson Sampling; WLAN; Channel Reuse; Clear Channel Assessment; Power Control

ACM Reference Format:
Anthony Bardou, Thomas Begin, and Anthony Busson. 2021. Improving the Spatial Reuse in IEEE 802.11ax WLANs: A Multi-Armed Bandit Approach. In *Proceedings of the 24th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '21), November 22–26, 2021, Alicante, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3479239.3485715

1 INTRODUCTION

Wireless Local Area Networks (WLANs) are overwhelmingly based on the IEEE 802.11 standard [1], which is commercially known as Wi-Fi. WLANs may be composed of one or multiple Access Points (APs) that serve as wireless gateways for the stations (STAs). In many urban places such as medium or large enterprises, universities, train stations, airports, and shopping centers, a large number of APs are deployed to form a very dense WLAN capable of ensuring proper radio coverage, increasing the physical transmission rates between STAs and APs, and supporting an important traffic load. WLANs composed of multiple APs are typically managed by a central controller to ease their management and the configuration of their APs. This facilitates the application of a consistent configuration to obtain a homogeneous setting (in terms of security for instance) and to optimize the WLAN performance.

WLANs' performance are sometimes regarded as relatively low, uncertain and uneven across the STAs. These issues mainly relate to the scarcity, time-varying nature, and non-exclusivity use of the radio channel as communication resource. In today's WLANs, the controller can attempt to mitigate these issues by staggering adjacent APs on different, non-overlapping radio channels. An efficient channel assignment can enable multiple APs to transmit simultaneously and successfully (either because they use different channels, or, because although using the same radio channel, their distance leads to a negligible amount of mutual interference) thereby favoring an efficient spatial reuse of radio channels. Unfortunately, the number of radio channels is limited and any channel assignment strategy will find its limitations when the density of APs is high.

The 802.11ax standard, which was approved in February 2021 and is marketed as Wi-Fi 6, introduces a new feature to further improve the spatial reuse of radio channels. Two key parameters, namely the transmission power and the sensitivity threshold referred to as `TX_PWR` and `OBSS/PD` respectively, have become tunable and can be set independently for each 802.11ax device. The setting of these parameters can have a profound impact on spatial reuse and thus on WLANs performance. However, no algorithms were included in the standard and it is up to the manufacturer or to the WLAN controller to decide how these parameters should be set.

Setting these two parameters is a difficult problem for three main reasons. First, any efficient setting of these parameters is tightly linked to the WLAN topology (i.e., arrangements of APs and STAs) and will likely be counterproductive for another WLAN. Second, the problem suffers from the curse of dimensionality. Because each AP has 2 parameters that can take 21 different values each, the number of possible network configurations grows in $O(2^{N_A})$ where N_A denotes the number of APs in the WLAN. This exponential growth of the size of the state space precludes the use of a brute-force approach even for a medium-sized WLAN. Third, the configuration

MSWiM '21, November 22–26, 2021, Alicante, Spain

ACM ISBN 978-1-4503-9077-4/21/11…$15.00
https://doi.org/10.1145/3479239.3485715

of these parameters must be found relatively quickly and without any noticeable disruption of service for the STAs. However, it is worth mentioning that the WLAN controller can perform configuration tests, and observe the resulting WLAN behavior on the STAs performance. This paves the way for the use of data-driven approaches such as reinforcement learning techniques to perform the search for an efficient setting of the WLAN parameters.

In this paper, our objective is to improve the spatial reuse of radio channels of WLANs by modifying the setting of the TX_PWR and OBSS/PD parameters of each AP. We cast the search for this parameter setting as a Multi-Armed Bandit (MAB) problem to which we propose a fast and efficient solution. Our solution can significantly improve the behavior of WLANs, leading in particular to a fairer share of the radio channel among the STAs. More precisely, our contributions are as follows:

- The efficiency of any reinforcement solution heavily relies on the definition of its reward function. We devised a reward function that accounts for the potential issues of WLANs (unfairness, starvations) and reflects the overall goodness of a network configuration from the standpoint of a network administrator.
- While a uniform sampling may initially appear a natural choice to explore the space of network configurations, we opt for a Gaussian mixture approach. This choice leverages a certain degree of smoothness in the rewards of similar network configurations and contributes to ensuring seamless and uninterrupted connectivity to the STAs.
- Using several WLAN scenarios run on a realistic network simulator, namely ns-3, we show the superiority of our approach at addressing the spatial reuse problem over traditional ways of performing the sampling and optimization steps within the MAB framework.

2 STATE OF THE ART

The literature related to the spatial reuse of radio channel can be classified into four major categories based on whether papers address the issue of channel allocation or the tuning of the TX_PWR and OBSS/PD parameters, and on whether the proposed solutions are based on analytical modeling, or conversely, mostly data-driven approaches. In this section, we review the literature associated with each group.

Allocating the radio channels

The first and foremost way of improving the spatial reuse of radio channels in an 802.11-based WLAN is obviously to allocate the same channel to multiple of its APs. Indeed, provided that two APs do not sense each other, they can then transmit at the same time without any risk of interfering, nor any need to share the communication resource materialized by the radio channel. The search for optimized solutions when multiple radio channel allocations exist is known as the channel allocation (CA) problem. Existing solutions to this problem are either model-based algorithms where an analytical model of the WLAN helps evaluate the quality of an allocation, or data-driven solutions where allocations are appraised through real measurements.

In model-based solutions, the CA problem can be tackled as a coloring problem with specific constraints. In the first generations of the 802.11 standard, channels had a fixed size of 20MHz. For example, a centralized algorithm has been proposed in [15]. Conflicts between APs are represented by a graph, and the solution aims to allocate different channels/colors to APs that are adjacent in this graph. The algorithm was evaluated using a real testbed. With the recent amendments to the 802.11 standard, several 20MHz channels can be aggregated into a 40, 80, or 160MHz channel. This technique known as channel bonding (CB) hardens the CA problem as the number of possible allocations increases significantly. In [9], a distributed algorithm named SA (Spectrum Assignment for WLAN) is formulated as an optimization problem. For a given topology, the algorithm aims at minimizing interference between APs while taking into account the preferences of APs for certain channel width. Another model-based approach is investigated in [11] where the analytical model accounts for both collisions and interference. In the case of an 802.11ac-based WLAN where the objective is to maximize the throughput for a given traffic demand, the authors approached the CB problem as an optimization problem whose solution is found through a genetic algorithm.

To be effective, model-based approaches require vast pieces of knowledge on the WLAN (topology, traffic, radio propagation, parameter setting, etc.), accurate analytical estimates of the network performance, and a relatively simple derivation. Unfortunately, it is often hard to meet these requirements and modeling approaches may be regarded as too inaccurate to handle the CA problem. On the other hand, data-driven approaches based on measurements collected from different WLAN configurations are intrinsically free of these constraints and thus appear promising. Some rely on heuristics (e.g., [4, 12]) while others make use of machine learning techniques (e.g., [13]). In [12], the authors propose a set of decentralized algorithms to allocate 20MHz channels to APs in order to efficiently reuse radio channel thereby increasing the overall network capacity. These algorithms are based on local measurements where, iteratively, each AP selects a channel with a certain probability, measures the channel performance for a certain period, and then adapts the probabilities for this channel accordingly. In [4], the proposed 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 efforts on other channels. In [13], the authors used a reinforcement learning algorithm to explore in real-time new configurations and to exploit the ones that offer better performance. The authors use a MAB approach with the Thompson sampling algorithm to select the new configurations to evaluate.

Tuning the TX_PWR and OBSS/PD parameters

Tuning the TX_PWR and OBSS/PD parameters (related to the transmission power and to the sensitivity threshold of nodes, respectively) is a secondary means (beyond CA) to improve the spatial reuse of a radio channel. Pioneering efforts were made in 2004 with [28] in which the authors present an analytical model for deriving

the optimal sensitivity threshold in a Wi-Fi mesh network. The physical carrier sensing threshold is tuned dynamically on each node as a function of the channel conditions. In [27], it is the transmission power that is tuned to increase the throughput and minimize the communication energy consumption. However, it is only in 2021 with the 802.11ax amendment that IEEE officially introduced the adaptation of the TX_PWR and OBSS/PD parameters thereby setting the technological context and constraints. In practice, the large number of parameters and the complexity of the physical layer in a radio environment hinders the use of such analytical model-based solutions. Instead, measurement-based techniques appear as natural candidates to this adaptation problem.

Practical approaches have been proposed in [17] and [18] to adapt the values of TX_PWR and OBSS/PD. In [17], the authors present a relatively simple way of dynamically tuning these two parameters. Using the Expected Transmission Count (ETX) value, their algorithm estimates a new value for TX_PWR as well as for OBSS/PD. In [18], a distributed solution aims to adapt dynamically the OBSS/PD as a function of the received signal strength. More precisely, the difference of signal strength between the frames in reception and the interfering frames (from other APs) is used to set a new value for OBSS/PD. The authors can control the likeliness of concurrent transmissions and thereby the level of "aggressiveness" in the selected configuration using an internal parameter of the algorithm.

Only a couple of works have proposed methods inspired by machine learning techniques to address the issue of tuning the TX_PWR and OBSS/PD parameters. In [25, 26], the authors formalize the problem of allocating the radio channel of APs and setting their TX_PWR and OBSS/PD parameters as a MAB problem. In both cases, the MAB algorithm is applied at each AP in a distributed way. The two solutions mostly differ in terms of their reward definition. In the first solution, the reward at each AP corresponds to its throughput, which can be described as a "selfish" solution since each AP tries to optimize its own reward independently of the other nodes. Conversely, the second reward revolves around a max-min function of the throughputs of the current AP and of its direct neighbors (set of nodes for which the current AP senses traffic). Using a home-made simulator, the authors show that their solution significantly outperforms the default configuration of the WLAN and that the selfish reward may lead to unfair situations between APs or STAs.

To summarize, only a limited number of studies have tackled the issue of setting the TX_PWR and OBSS/PD parameter in an attempt to increase the spatial reuse of radio channel for WLANs. Data-driven approaches such as reinforcement learning techniques appear well suited to deal with the intrinsic complexities of this issue. Unlike a couple of previous works that proposed distributed approaches wherein each AP sets its parameters based on its knowledge [25, 26], we introduce a centralized solution in which the WLAN controller configures the parameters of the APs composing its fleet. We propose a novel definition for the reward function specifically tailored to WLAN performance. Additionally, we devise a Gaussian mixture-based approach to explore the quasi-infinite state space of configurations. Finally, to the best of our knowledge, we are the first to use the popular open source network simulator ns-3, which includes a realistic representation of the physical, link, network, transport and application layers, to show the efficiency of our solution at setting the TX_PWR and OBSS/PD parameters. We believe that the use of this well-established simulator strengthens the validation of our solution.

3 WLAN UNDER STUDY

We consider a WLAN comprising multiple APs and stationary STAs as well as a controller that configures and manages the WLAN. STAs are associated to the AP with the strongest signal strength. To access the radio channel, APs and STAs use a listen-before-talk scheme referred to as carrier-sense multiple access with congestion avoidance (CSMA/CA) and accomplished by the distributed coordination function (DCF) in the 802.11 standards. DCF requires each node (AP and STA) willing to transmit to first sense the radio channel state for a short period of time. If the channel is sensed busy, the node will defer its transmission for a random period of time called backoff. If the channel is sensed idle (or after the backoff timer has come to zero), the node is allowed to transmit its frame. For more details, we refer the interested reader to [7].

The 802.11 standards rely on a clear channel assessment (CCA) function to indicate if the radio channel is perceived as busy or idle. Although other options are made possible, CCA is most often performed by comparing the power of the received signal (in dBm) to a given ceiling threshold often referred to as sensitivity and denoted by OBSS/PD. If the former exceeds the latter, the radio channel is considered busy. Otherwise, it is detected as idle. Until the recent release of 802.11ax, the OBSS/PD was set to a constant value (e.g., -82dBm for 802.11n). Analogously, the transmission power denoted by TX_PWR, which deviates from the received signal power due to the path loss and shadowing effects, was also constant and often set to 20dBm [8]. The latest amendment of 802.11, namely 802.11ax, enables the values of TX_PWR and OBSS/PD to be dynamically changed within certain ranges (e.g., [10]). While TX_PWR can take all values in between 1 and 21dBm, OBSS/PD can vary from -82 to -62dBm provided the two parameters meet relation (1), given by [2]. In our case, we assume that the WLAN controller is able to set the TX_PWR and OBSS/PD values for each AP.

$$\text{OBSS/PD} \leq \max(-82, \min(-62, -82 + (20 - \text{TX\_PWR}))). \quad (1)$$

A node is said to be in conflict with another if the former is made unable to transmit (due to the outcome of its CCA function) when the latter is currently transmitting. Conflicts between APs heavily influences the performance of a WLAN. They reduce channel interference and the probability of colliding frames but they also tend to limit the number of simultaneous transmissions in a WLAN and hence the spatial reuse of a radio channel. Due to the importance of conflicts in the understanding of a WLAN behavior, it is a common practice to represent WLANs by their conflict graph between APs (e.g., [6, 15, 21]). Note that conflicts of STAs are typically not represented in conflict graphs as the vast majority of traffic in WLANs is downstream (STAs typically generate at least an order of magnitude less traffic than APs). Figure 1 shows two conflict graphs associated with the same WLAN but with different settings of their AP's TX_PWR and OBSS/PD parameters. We can see the corresponding conflicts between APs when all APs have the same setting (default value). Conversely, when APs have different settings for their TX_PWR and OBSS/PD parameters, we observe that,

with this particular setting, the number of conflicts between APs, which are no more symmetrical, has significantly decreased.

Several performance metrics are worth of interest to evaluate the efficiency of a WLAN at providing wireless access to its STAs. First, the aggregate throughput (also known as system throughput) represents the sum of the throughputs of all individual STAs in the WLAN. Second, the fairness in the distribution of access to the radio channel among STAs is another critical factor. Measures of fairness such as Jain's index or proportional fairness (PF), based on the individual throughputs of all STAs, are common means to determine whether certain STAs are receiving a disproportionate share of the radio resource at the expense of other STAs. Indeed, certain STAs may have a very limited access to the radio channel due to their position in the conflict graph. These STAs are said to be in starvation of throughput and they represent a major issue for network administrators. In this paper, a STA is considered to be starving if it cannot obtain at least a given percentage α, say 10%, of the throughput they would have in the absence of other STAs. Third, the frame error rate (FER) of each STA, which indicates the percentage of frames lost due to collisions and poor channel condition, can also be worth of interest to network administrators. As discussed earlier in this section, the setting of `TX_PWR` and `OBSS/PD` on each AP can significantly change these performance metrics. For instance, Figure 1 shows the conflict graph associated with a WLAN for two different parameter settings. While the default setting (see Figure 1) leads to an aggregate throughput of 600Mbps, a Jain's index of 0.42, and a number of starving STAs at 8, a more appropriate setting of these parameters (illustrated again by Figure 1) can shift these values to 900Mbps, 0.55, and 2, respectively.

4 PROPOSED SOLUTION

In our solution, we search for a correct network configuration using an agent located in the network controller. We assume that regular reports on the individual throughputs of STAs are forwarded to the WLAN controller, which implements a MAB approach. In this framework, each network configuration c_i ($i > 1$) represents an arm that the agent can pull (thus performing a trial) to obtain a reward r_i^k with k designating the trial index. Note that the rewards of an arm i are drawn from a probability distribution $\mathcal{D}(\theta_i)$ whose parameters θ_i are unknown from the agent but invariant in time since the network under study is assumed to be stationary. In this section, we present an efficient strategy $\pi(k)$, which determines which arm to pull (WLAN configuration to test) at each trial (time step) k.

Given the exponential growth of the cardinality of the configuration space C with the number N_A of APs (bounded by 21^{2N_A} since an AP has two tunable parameters of 21 values each), our problem is more precisely framed as an Infinitely Many-Armed Bandit (IMAB). Thus, in practice, the network controller cannot explore the whole set of arms in a reasonable amount of time and must instead work on a subset of C, referred to as the reservoir. In fact, the optimal arm is likely to not even be considered during the search process.

Table 1 summarizes the principal notation used in our proposed solution together with their numerical values chosen empirically for our simulations.

Table 1: Principal notation for the proposed method and their corresponding values in the simulations.

Parameter	Value	Description
C	Relative to topology	Configuration space
N_A	Relative to topology	Number of APs
N_S	Relative to topology	Number of STAs
T_i	Relative to topology	Throughput of STA i
T_i^A	Relative to topology	Attainable throughput of STA i
T^-	Relative to topology	STAs in starvation situation
T^+	Relative to topology	STAs not in starvation situation
α	0.1	Starvation threshold
ϵ	0.1	Exploration rate for strategies
n	2	Sample size in Algorithm 1
K	6	Number of Gaussians in Algorithm 2
δ	$\frac{1}{1+N_s}$	Hypothesis parameter in Algorithm 2

4.1 Reward function

In a Reinforcement Learning (RL) problem, the choice of the reward function is a critical step and its definition can deeply influence the outcome of the optimization process. In the case of a WLAN, the reward function aims at quantifying the quality of a network configuration. However, as discussed in Section 3, there are several performance metrics to assess the quality of a WLAN, and thus different ways of combining them. From the standpoint of a network administrator, a WLAN configuration is considered favorable if it ensures a fair share of throughput among the APs and STAs. More precisely, we enumerate, by order of importance, three criteria to take into account: (i) the number of STAs that are starving for throughput should be minimized, (ii) the fairness between STAs should be maximized, (iii) the aggregate throughput of the network should be maximized.

Satisfying both criteria (ii) and (iii) at the same time is challenging. Indeed, in most topologies, increasing fairness between throughputs of STAs is made at the expense of a lower aggregate throughput. Conversely, increasing the network aggregate throughput often implies a decrease in fairness. In order to reach a natural trade-off between those two metrics, we build our reward function on a normalized version of the proportional fairness (PF) of the station throughputs as given by Equation 2. Note that in Equation 2, the STAs' throughputs $T_i, i \in [1, N_S]$ are normalized by their attainable throughputs $T_i^A, i \in [1, N_S]$. T_i^A is simply defined as the throughput that the STA i would obtain in the absence of all other stations. Hence, T_i^A can be seen as an upper bound for T_i. Then, the normalized throughputs are multiplied with each other to obtain a ratio belonging to $[0, 1]$ that represents the quality of the compromise found between criteria (ii) and (iii).

$$\mathrm{PF}(T, T^A) = \prod_{i=1}^{N_S} \frac{T_i}{T_i^A} \tag{2}$$

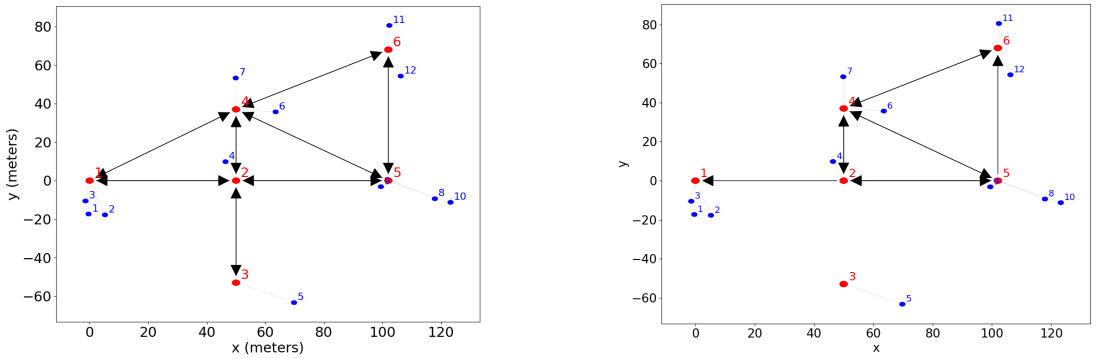

Figure 1: Example of two conflict graphs resulting from different settings of `TX_PWR` and `OBSS/PD` for a same WLAN. Left: default configuration (`TX_PWR`, `OBSS/PD`) = (20,-82) dBm for all APs. Right: (`TX_PWR`, `OBSS/PD`) = (15,-81), (18,-80), (17,-79), (19,-82), (20,-82), (19,-82) dBm.

We account for criterion (i) by ensuring that any network configuration with a higher number of stations in starvation situations than another configuration obtains a lower reward value. In our case, we consider that STA i is starving for throughput whenever its throughput T_i is less than a given fraction, denoted by $\alpha \in [0, 1]$, of its attainable throughput T_i^A. For this work, we chose $\alpha = 0.1$. Therefore, a station having a throughput less than 10% of its attainable throughput is said to be in starvation. Having defined the notion of starvation, we regroup STAs in starvation (those whose $T_i < \alpha T_i^A$) in a set T^- while the others are placed in a set T^+. Then, we compute the PF for each subset T^+ and T^-, we normalize them using their upper bounds (αT_j^A for T^- and T_j^A for T^+), and we combine them using Equation 3 to obtain our reward function. Note that this definition forces our reward to evolve in disjoint intervals in $[0, 1]$, the selected one depending on the number of STAs in starvation, as depicted by Figure 2.

$$r_i^k = \frac{|T^-| \prod_{j \in |T^-|} \frac{T_j^-}{\alpha T_j^A} + |T^+| \left(N_S + \prod_{j \in |T^+|} \frac{T_j^+}{T_j^A} \right)}{N_S(N_S + 1)} \tag{3}$$

To measure the quality of a given strategy $\pi(k)$, we use the cumulative regret, which is the standard metric used in MAB problems. With μ^* denoting the best expected reward (i.e., $\mu^* = \max_c \mathbb{E}[r|c]$), the cumulative regret $R_n(\pi)$ on strategy π after n actions (or trials) taken by the agent, is defined by Equation 4.

$$R_n(\pi) = n\mu^* - \sum_{k=1}^{n} r_{\pi(k)}^k \tag{4}$$

A common practice to circumvent the infinite number of arms in an IMAB problem consists of restricting the exploration to a limited subset of solutions composed of random arms that constitute the reservoir. Typically, the selected arms are drawn uniformly from the whole set of arms (e.g., [24], [5], [14]). In our case, this approach is not suitable as the vast majority of network configurations lead to poor solutions so that the reservoir would therefore consist only of unsuitable solutions. However, unlike a typical IMAB problem, in which no hypothesis can be made on the relationship between the arms and their rewards, in our case, two neighbor network configurations are likely to have similar rewards. To exploit this similarity between neighbor configurations, we consider our configuration space as a normed space with $||\cdot||$ the L1-norm and we assume that the property given by Equation 5, which relates the spaces of arms and rewards, is verified. Although not always true, this property enables us to leverage the information collected on former trials to guide the sampling of new configurations, mostly in the neighborhood of already good configurations.

$$\forall c_i, c_j \in C, \exists \delta > 0, ||c_i - c_j|| = 1 \implies |r_i^{\cdot} - r_j^{\cdot}| < \delta \tag{5}$$

Therefore, our problem breaks down into two subproblems that must be solved concurrently: (i) sampling promising configurations, and (ii) identifying the best arm among those sampled and pulling it as much as possible. An agent called the sampler is in charge of the first task while another agent known as the optimizer accounts for the second task. Figure 3 summarizes the main principles of our solution. The remainder this section is devoted to the definition of the optimizer and the sampler agents.

4.2 Optimizer

The role of the optimizer is to quickly identify the best network configuration (a.k.a. $\text{argmax}_c \mathbb{E}[r|c]$) among the current reservoir of network configurations and to use it most often. In our case, the reservoir is initialized with the default configuration of APs and progressively filled with new configurations proposed by the sampler. Algorithm 1 describes the behavior of the optimizer agent. The algorithm has two main parameters: the exploration rate ϵ, which decides how often configurations out of the reservoir are tested, and the sample size n, which determines how often reward estimates of a given configuration are updated.

Algorithm 1 is based on Thompson Sampling (TS) [23], which achieves an optimal regret bound [19, 20] and derives from Bayesian principles. In the previous works [25, 26] wherein TS was used for the sake of spatial reuse of WLANs, the authors assume that $(r_i|c_i) \sim \mathcal{N}(\mu_i, 1)$, with a Gaussian prior for μ_i initialized at $\mathcal{N}(0, 1)$. Then, the prior of μ_i at step k is given by $\mathcal{N}\left(\hat{\mu}_i^k, \frac{1}{n_i^k+1}\right)$, where

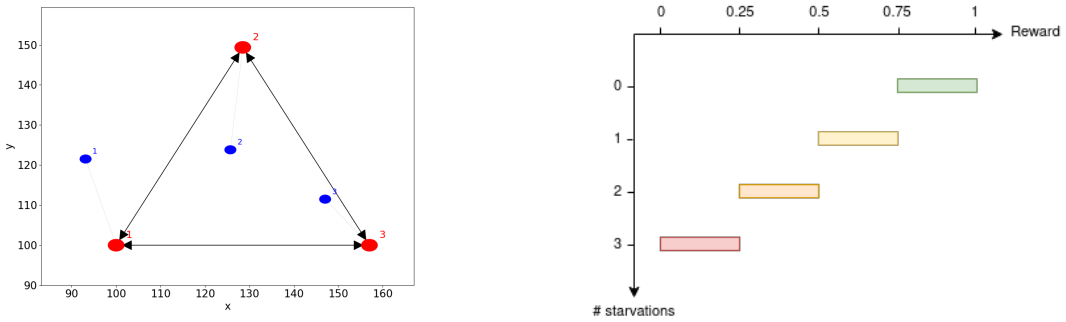

Figure 2: On the left, a simple topology composed of 3 APs and 3 STAs. On the right, the corresponding reward intervals, containing increasing values as the number of STAs in starvation decreases.

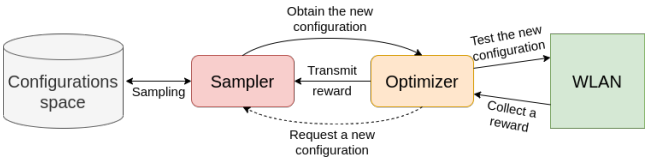

Figure 3: Outline of our solution: The optimizer requests a new configuration to the sampler, which selects and returns it to the optimizer. The optimizer tests this configuration on the real environment, obtains a reward in return, and forwards this reward to the sampler so that both agents can update their internal state.

$\hat{\mu}_i^k = \frac{\sum_{w=1:i(w)=i}^{k-1} r_i^k}{n_i^k+1}$ ([26]) and n_i^k denotes the number of times the configuration c_i has been tested after k steps. In our solution, we make no assumptions regarding the actual value of the variance of r_i. Similarly to the mean value, the variance is progressively estimated. We assume $(r_i|c_i) \sim \mathcal{N}\left(\mu_i, \sigma_i^2\right)$ and choose normal-gamma priors for both the mean μ_i and the precision σ_i^{-2}: $\left(\mu_i, \sigma_i^{-2}\right) \sim \text{NormalGamma}\left(\hat{\mu}_i^k, \hat{\lambda}_i^k, \hat{\alpha}_i^k, \hat{\beta}_i^k\right)$. Note that a normal-gamma distribution implies that the precision (inverse to the variance) has a Gamma distribution and that the mean, once the precision is known, has a Normal distribution. Therefore, the priors at step k for μ_i are given by $(\mu_i|G) \sim \mathcal{N}\left(\hat{\mu}_i^k, \left(\hat{\lambda}_i^k G\right)^{-1}\right)$, where $G \sim \Gamma\left(\hat{\alpha}_i^k, \hat{\beta}_i^k\right)$. For a sample X_i of size n, mean $\bar{x}_i$, and variance s_i, standard calculations demonstrate that $\left(\mu_i, \sigma_i^{-2}|X_i\right) \sim \text{NormalGamma}\left(\hat{\mu}_i^{k+1}, \hat{\lambda}_i^{k+1}, \hat{\alpha}_i^{k+1}, \hat{\beta}_i^{k+1}\right)$, with Equation 6 describing how to update our parameters. These updates enable our optimizer to incorporate new measures on the network configuration into its reward estimations.

$$\begin{pmatrix} \hat{\mu}_i^{k+1} \\ \hat{\lambda}_i^{k+1} \\ \hat{\alpha}_i^{k+1} \\ \hat{\beta}_i^{k+1} \end{pmatrix} = \begin{pmatrix} \frac{\hat{\lambda}_i^k \hat{\mu}_i^k + n\bar{x}_i}{\hat{\lambda}_i^k + n} \\ \hat{\lambda}_i^k + n \\ \hat{\alpha}_i^k + \frac{n}{2} \\ \hat{\beta}_i^k + \frac{1}{2}\left(ns_i + \frac{\hat{\lambda}_i^k n\left(\bar{x}_i - \hat{\mu}_i^k\right)^2}{\hat{\lambda}_i^k + n}\right) \end{pmatrix} \tag{6}$$

Algorithm 1 Optimizer algorithm

Input: sample size n, exploration rate ϵ

Init reservoir E with $\emptyset$
Init step counter k with 0
loop
  if $E = \emptyset$ **or** rand() $< \epsilon$ **then**
    Get a new configuration c_i using the sampler
    Test c_i n times on the environment and collect rewards in X_i
    $k \leftarrow k + n$
    $\left(\mu_i^k, \lambda_i^k, \alpha_i^k, \beta_i^k\right) \leftarrow \left(\bar{X}_i, n, \frac{n}{2}, \frac{nVar(X_i)}{2}\right)$
    $X_i \leftarrow \emptyset$
    Add c_i to reservoir E
  else
    for c_i **in** E **do**
      Sample g_i from $\Gamma\left(\alpha_i^k, \beta_i^k\right)$
      Sample μ_i from $\mathcal{N}\left(\mu_i^k, \left(\lambda_i^k g_i\right)^{-1}\right)$
    end for
    $j \leftarrow \text{argmax}_i\, \mu_i$
    Test c_j on the environment and add reward to X_j
    $k \leftarrow k + 1$
    if $|X_j| = n$ **then**
      Update prior parameters $\left(\mu_j^k, \lambda_j^k, \alpha_j^k, \beta_j^k\right)$ according to Equation 6
      $X_j \leftarrow \emptyset$
    end if
  end if
  Send tests and rewards to the sampler algorithm
end loop

4.3 Sampler

The role of the sampler is to explore the configuration space C, and to yield promising configurations when requested to by the optimizer. The exploration process is given by Algorithm 2.

Algorithm 2 Sampler algorithm

Input: K number of Gaussians, δ target parameter

if first call **then**
  Init G with $\left\{\left((-82, 20, \cdots, -82, 20), \left(\frac{1}{\dim C}\right)^2 I\right)\right\}$
  Init weights W with $\{1\}$
  Init history H with $\emptyset$
  Init tests counter k with 0
else
  Retrieve previously built G, W, H and k
  Add pairs (conf, rew) transmitted by the optimizer
end if
Sample a new configuration c from mixture (G, W)
Transmit c to the optimizer
$k \leftarrow k + 1$
if $k = \sum_{(\mu_i, \lambda_i I) \in G} \lambda_i \dim \mu_i$ **then**
  Reset G and W
  Find K (c_i, r_i) pairs in H with largest rewards
  $target \leftarrow \delta + \max_j r_j$
  for $i \leftarrow 0$ **to** K **do**
    Add $\left\{\left(c_i, \left(\frac{target - r_i}{\delta \dim C}\right)^2 I\right)\right\}$ to G
    Add r_i to W
  end for
end if

To efficiently sample new configurations in an infinite space, we build our sample distribution as a normalized sum of Gaussian distributions, which is known as a Gaussian Mixture (GM). Algorithm 2 constructs and updates this GM. Unlike uniform sampling, a GM-based sampling whose Gaussians are centered on the best-known configurations ensures that most of the new sampled configurations lye in the vicinity of the currently best-known configurations.

We allow a total of K Gaussian distributions in the mixture and we propose to define their centers as the K best configurations discovered so far, whose associated rewards are denoted by $r_1 \geq \cdots \geq r_K$. To sample in every direction without distinction, their covariance matrices will be scalar: $\Sigma_i = \lambda_i I$, $\lambda_i \in \mathbb{R}^+$. In order to find an adequate value of λ_i, we consider the hypothesis made on Equation 5. If Equation 5 is true, then $\forall c_i, c_j \in C, \exists \delta > 0, ||c_i - c_j|| = x \implies |r_i^{\cdot} - r_j^{\cdot}| < x\delta$. Targeting a new configuration with a reward of $r_1 + \delta$ for the next sample, and considering the i-th Gaussian centered on c_i with an average reward of r_i, we need to sample a new configuration c so that $||c_i - c|| \geq \frac{r_1 + \delta - r_i}{\delta}$. One way to sample configurations which are, on average, away from c_i by this distance is to set $\lambda_i = \left(\frac{r_1 + \delta - r_i}{\dim C * \delta}\right)^2$. Thus, parameterized by K and δ, our sampling strategy defines a mixture of K Gaussians centered on the K best configurations discovered so far; the i-th Gaussian being defined by $\mathcal{N}\left(c_i, \left(\frac{r_1 + \delta - r_i}{\dim C * \delta}\right)^2 I\right)$.

5 NUMERICAL RESULTS

5.1 Experimental settings

To evaluate the efficiency of our solution at improving the spatial reuse of a WLAN, we implemented it in the realistic discrete-event simulator ns-3 [16] and explored its performance against those of other existing strategies. The ns-3 code implementing our solution, the other strategies, as well as the considered topologies are available for download at [3].

Table 2: ns-3 parameters.

Parameter	Value
ns-3 version	3.31
Number of repetitions	25
Simulation duration	120 s
Test duration	50 ms
Packet size	1,464 Bytes
Frequency band	5 GHz
A-MDPU Aggregation	4
Path loss	`LogDistancePropagationLossModel`
MCS Control	VhtMcs0
MCS Data	VhtMcs4

In addition to our solution described in Section 4, we consider three other strategies that we also implemented in the simulator ns-3. First, we include the classical ϵ-greedy strategy [22], which, at each step, either tests a random configuration with probability ϵ, or chooses the best configuration so far with probability $1 - \epsilon$. In the remainder of this section, we use ϵ-`GREEDY` to denote this simple strategy and we use $\epsilon = 0.1$. Second, we implement a solution based on Thompson Sampling but using the priors proposed by [25]. We use `TS` to refer to this solution. Note that the authors of [25] proposed the use of `TS` in a different technological context than ours (distributed version) nullifying any comparison beyond our context. The third strategy is a modified version of `TS` wherein the sampler is replaced with ours based on Gaussian Mixture (Section 4.3). We refer to this last strategy as `GM-TS`. ϵ-`GREEDY` and `TS` strategies are using uniform sampling to discover new configurations. Therefore, comparing `TS` and `GM-TS` allows us to quantify the benefits brought by our sampling algorithm while comparing `GM-TS` and our solution (sampler & optimizer) highlights the benefits of our optimizer. Table 1 indicates the parameter values used for all the considered strategies.

In each of our experiments, the simulation runs last for a total of 120 seconds of simulated time. For the sake of accuracy, each simulation was replicated with 25 independent repetitions. The duration of a test, corresponding to the length of a trial for our solution, was set to 50 msec. Therefore, 2,400 optimization steps can be performed before the simulation ends. Because we replicate 25 times each simulation, we obtain a matrix of 25x2,400 measures for each network performance metric. In order to provide a clear visualization of this large set of data, we chose to plot the median of the metric at each optimization step, framed by its first and third quartiles. Finally, we applied an exponential moving average (EMA) with a parameter of 0.04 to the three considered quartiles, so we can notice the trends caused by the optimization. This kind of visualization gives us an insight not only into the final performance of each strategy but also on its performance during the whole optimization process. The remainder of the ns-3 simulation parameters is given by Table 2. For the sake of comparison, all strategies are evaluated using the same simulation parameters as well as the same reward function.

We present three examples out of the many we investigated corresponding to the network topologies **T1**, **T2** and **T3** depicted in Figure 4. Each of them may correspond to a typical dense WLAN deployment. Topologies **T1** and **T2** are both composed of 6 APs, each being associated with two or three STAs. As for the topology **T3**, it is composed of 10 APs and 25 STAs. **T3** is particularly dense with an average of 5.6 conflicts per AP (when configured with the default setting of TX_PWR and OBSS/PD), and will allow us to test our solution on a larger, denser WLAN deployment. Note that the number of APs here refers to APs belonging to the same WLAN and set on the same radio channel. Given the number of independent channels (3 in 2.4GHz and 23 in 5GHz in many countries), topologies like **T1**, **T2** and **T3** could actually correspond to WLANs comprising dozens of APs.

5.2 Simulation results

We start our performance analysis by studying the evolution of the number of starving STAs with each strategy. Recall that starving STAs represent a major issue for WLANs and an efficient WLAN configuration should be able to remove as many starving STAs as possible. Figure 5 shows the corresponding results delivered by the simulator ns-3. We notice that initially, with the default setting of TX_PWR and OBSS/PD, the number of STAs in starvation is in average at 10 for **T1**, 7 for **T2** and 18 for **T3**. However, with the exception of TS, all strategies manage to rapidly reduce the number of starving STAs across the three examples. The results also show that GM-TS significantly outperforms TS suggesting the importance of the sampling process in the overall optimization. Finally, Figure 5 indicates that our solution leads to the removal of a proportion of starving STAs ranging between 10 and 65% depending on the considered topology.

To further illustrate the gain that a better setting of the TX_PWR and OBSS/PD parameter values can have on the network, we represent in Figure 6 the throughputs of each STA for both the default configuration of 802.11ax and the one found by our solution on **T2**. Figure 6 shows that all STAs achieve higher throughputs when using the configuration found by our solution. More importantly, as pointed by Figure 6, our solution enables most STAs to operate above the starvation threshold and only 3 of them (STA 4, STA 8 and STA 9) are occasionally experiencing starvation of throughput. Figure 6 shows that, in the case of the default configuration, most STAs are at least periodically experiencing starvation of throughput. Similar results (not presented in this paper) were obtained for **T1** and **T3**.

We now explore the influence of our solution over the fairness that reflects how uniformly the throughputs are distributed among the STAs. For that purpose, we use Jain's index that tends to be negatively correlated to the number of STAs in starvation in the network. Figure 7 represents the corresponding results for each topology. We observe that our solution leads to a quick increase of the fairness during the search by 30 to 50 points when compared to TS and brings a substantial gain from the fairness associated with ϵ-GREEDY.

For the sake of completeness, we study the influence of all strategies over the aggregate throughput, defined as the sum of the STAs throughputs (see Section 3). Figure 8 reports the corresponding results. We observe that out of the 4 considered strategies, ϵ-GREEDY is the one that leads to the largest improvement in terms of aggregate throughput. ϵ-GREEDY performs respectively around 37%, 14% and 7% better than our method on the topologies **T1**, **T2** and **T3**. However, keep in mind that maximizing the aggregate throughput is only a secondary objective in a WLAN and that it is unfortunately often done at the expense of fairness and the number of starving STAs (as shown by Figures 5 and 7). Figure 8 shows that for topology **T1** our proposed solution maintains the aggregate throughput near its original value obtained with the default setting of TX_PWR and OBSS/PD. For topologies **T2** and **T3**, our solution was able to significantly increase the aggregate throughput. Overall, these results indicate that there is no downside in the aggregate throughput to the significant benefits brought by our solution.

Lastly, we assess the performance of our solution with regards to the standard measure of quality in MAB problems, namely the cumulative regret, which represents the sum of differences between the maximum reward and the reward obtained at each trial (as defined by Equation 4). Figure 9 represents the cumulative regret obtained by each of the four strategies across the three topologies. This figure clearly shows that our solution is the one that provides the lowest cumulative regret on each topology during the whole optimization process. Furthermore, comparing the strategies TS and GM-TS shows the positive influence that the sampler can have while the comparison between the results of GM-TS and our solution points out the importance of the optimizer and of the priors in use in a Thompson Sampling approach.

Within 2,400 iterations, representing 120 seconds of simulated time and a very limited exploration in large state spaces, our solution was always able to significantly reduce the number of starving STAs and to increase fairness between the throughputs of STAs without decreasing the aggregate throughput of the WLAN. Note that better results may be achieved with longer simulations. Those improvements are obtained in less than 900 iterations representing 45 seconds for the smaller topologies **T1** and **T2**. We believe that these results demonstrate the capacity of a tailored MAB solution at improving the spatial reuse of radio channels in WLANs.

6 CONCLUSIONS

We have presented a new solution to improve the spatial radio reuse of 802.11ax-based WLANs by configuring two parameters at each AP, namely their transmission power and their sensitivity threshold. More precisely, we introduced a reward function specifically tailored to our purpose that quantifies the quality of a WLAN configuration. To help the exploration process at discovering promising configurations, we present a new way of sampling the state space that differs from uniform sampling, and of decoupling the search for the best configuration among those discovered so far and the discovery of new promising configurations. The obtained results on ns-3 demonstrate the large potential benefit brought by adapting the transmission power and sensitivity threshold of APs, as well as the ability of our solution to find an adequate configuration.

A natural follow-up to our work is to implement our solution in a WLAN to test our solution in a real-world environment. Other future works include the extension of our approach to a distributed

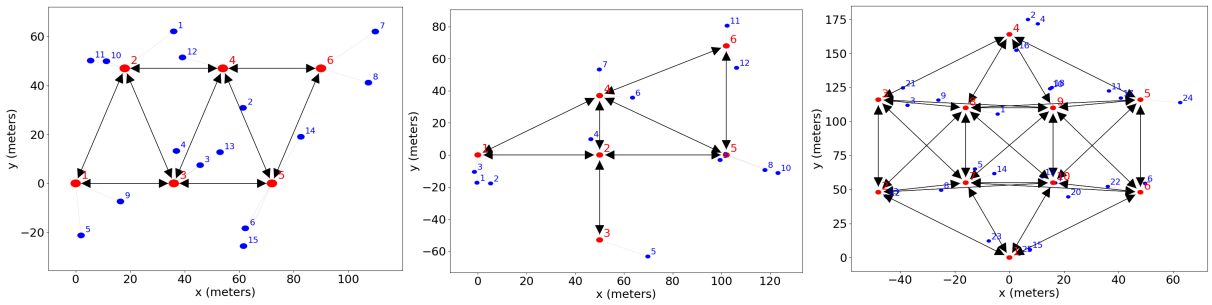

Figure 4: The three topologies T1, T2 and T3 used in the evaluation of our proposed method. The APs are shown with red circles while the conflicts between APs can be seen with black two-headed arrows. The STAs are represented with blue circles.

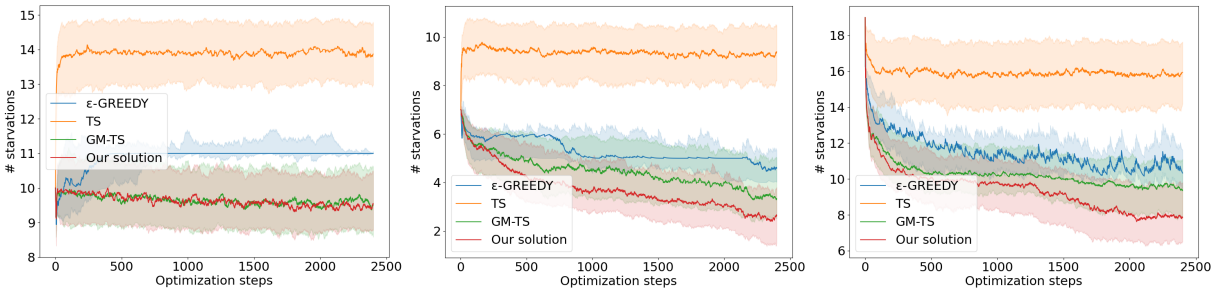

Figure 5: Evolution of the number of STAs starving of throughput for the four considered strategies on T1, T2 and T3.

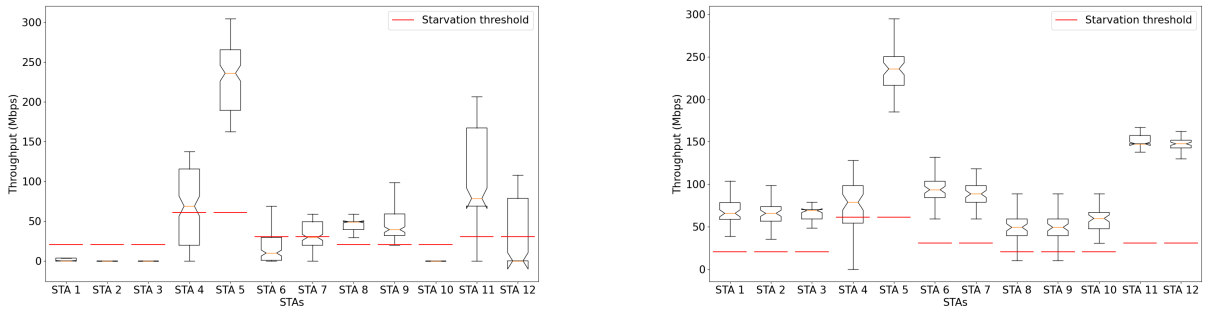

Figure 6: Throughputs obtained by STAs for the default 802.11ax configuration (left) and the configuration found by our algorithms (right) on T2. Each STA throughput distribution is shown as a boxplot, with a red horizontal bar designating the starvation threshold: if the throughput is below this bar, the STA is considered as starving.

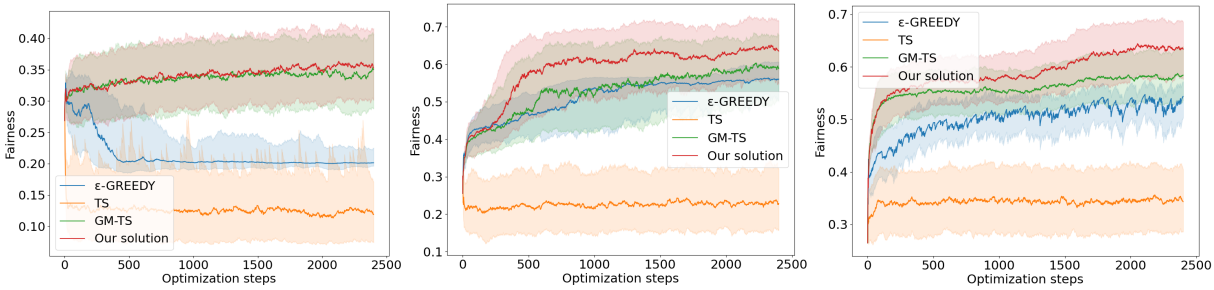

Figure 7: Evolution of fairness between the STA throughputs for the four considered strategies on T1, T2 and T3

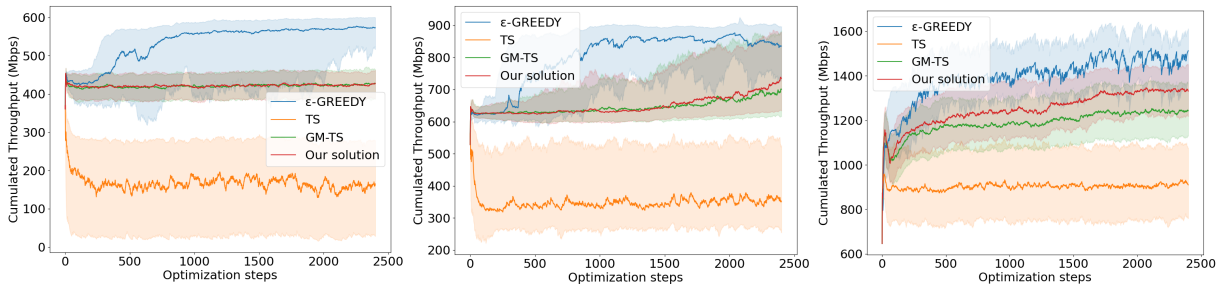

Figure 8: Evolution of the aggregate throughput for the four considered strategies on T1, T2 and T3.

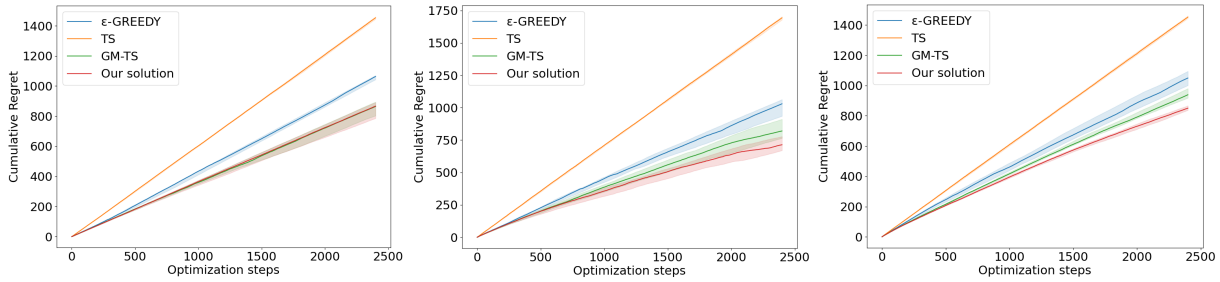

Figure 9: Evolution of the cumulative regrets for the four considered strategies on T1, T2 and T3.

context making it applicable to WLANs that do not include a network controller.

ACKNOWLEDGMENTS

This work was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX- 0007) operated by the French National Research Agency (ANR).

REFERENCES

[1] 2021. IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)* (2021).

[2] 2021. IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN. *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)* (2021), 1–767. https://doi.org/10.1109/IEEESTD.2021.9442429

[3] A. Bardou, T. Begin, and A. Busson. 2021. Online repository for code. https://github.com/abardou/IMAB-SR-STA-802.11ax.

[4] 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* (2020).

[5] Y. David and N. Shimkin. 2014. Infinitely Many-Armed Bandits with Unknown Value Distribution. (2014).

[6] M. Garetto, T. Salonidis, and E.W. Knightly. 2008. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. *IEEE/ACM Transactions on Networking* (2008).

[7] M. Gast. 2005. *802.11 wireless networks: the definitive guide.* O'Reilly Media, Inc.

[8] M. Gast. 2012. *802.11n: the survival guide.* O'Reilly Media, Inc.

[9] J. Herzen, R. Merz, and P. Thiran. 2013. Distributed spectrum assignment for home WLANs. In *IEEE INFOCOM.*

[10] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi. 2018. A tutorial on IEEE 802.11 ax high efficiency WLANs. *IEEE Communications Surveys & Tutorials* (2018).

[11] 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* (2019).

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

[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* (2020).

[14] Aziz M., Anderton J., Kaufmann E., and Aslam J. 2018. Pure Exploration in Infinitely-Armed Bandit Models with Fixed-Confidence. (2018).

[15] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. 2006. A Client-Driven Approach for Channel Management in Wireless LANs. In *IEEE INFOCOM.*

[16] ns3 2021. The Network Simulator ns-3. https://www.nsnam.org/.

[17] T. Ropitault and N. Golmie. 2017. ETP algorithm: Increasing spatial reuse in wireless LANs dense environment using ETX. In *IEEE PIMRC.*

[18] I. Selinis, K. Katsaros, S. Vahid, and R. Tafazolli. 2018. Control OBSS/PD Sensitivity Threshold for IEEE 802.11ax BSS Color. In *IEEE PIMRC.*

[19] A. Shipra and G. Navin. 2012. Analysis of Thompson Sampling for the Multi-Armed Bandit problem. (2012).

[20] A. Shipra and G. Navin. 2013. Further Optimal Regret Bounds for Thompson Sampling. (2013).

[21] M. Stojanova, T. Begin, and A. Busson. 2019. Conflict graph-based model for IEEE 802.11 networks: A Divide-and-Conquer approach. *Performance Evaluation* (2019).

[22] R. Sutton and A. Barto. 2018. *Reinforcement learning: An introduction.* MIT press.

[23] W. R. Thompson. 1933. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika* (1933).

[24] Y. Wang, J-Y. Audibert, and R. Munos. 2009. Algorithms for Infinitely Many-Armed Bandits. (2009).

[25] F. Wilhelmi, S. Barrachina-Muñoz, C. Cano, B. Bellalta, A. Jonsson, and G. Neu. 2018. Potential and Pitfalls of Multi-Armed Bandits for Decentralized Spatial Reuse in WLANs. *CoRR* (2018).

[26] F. Wilhelmi, C. Cano, G. Neu, B. Bellalta, A. Jonsson, and S. Barrachina-Muñoz. 2017. Collaborative Spatial Reuse in Wireless Networks via Selfish Multi-Armed Bandits. *CoRR* (2017).

[27] K. Youngsoo, Y. Jeonggyun, and C. Sunghyun. 2004. SP-TPC: a self-protective energy efficient communication strategy for IEEE 802.11 WLANs. In *IEEE VTC.*

[28] J. Zhu, X. Guo, L. Lily Yang, W. Steven Conner, S. Roy, and M. M. Hazra. 2004. Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks. *Wireless Communications and Mobile Computing* (2004).