Multi-Objective Surrogate Modeling for Real-Time Energy-Efficient Station Grouping in IEEE 802.11ah

Le Tian[a], Michael Mehari[b], Serena Santi[a], Steven Latré[a,b], Eli De Poorter[b], Jeroen Famaey[a]

[a] *University of Antwerp - imec, IDLab, Belgium*
[b] *Ghent University - imec, IDLab, Belgium*

Abstract

The Restricted Access Window (RAW) mechanism proposed by IEEE 802.11ah promises to address one of the major problems of the Internet of Things (IoT): high channel contention in large-scale densely deployed sensor networks. The RAW feature allows the Access Point (AP) to divide stations into different groups, with only the stations in the same group being allowed to access the channel simultaneously. Existing station grouping strategies only support homogeneous scenarios, where all sensor stations have the same fixed data transmission interval, modulation and coding scheme (MCS) and packet size. In this paper, we present two contributions to address this issue. First, we propose a surrogate modeling methodology to predict RAW performance for both throughput and energy efficiency given specific network conditions and RAW configuration parameters. The models are fast to train and can be solved in real-time. Second, we extend the original version of Model-Based RAW Optimization Algorithm (MoROA) which only focused on throughput. The extended MoROA uses the surrogate models to determine the optimal RAW configuration for both throughput and energy efficiency in real-time through multi-objective optimization. We compare the accuracy of our surrogate model to simulation results. Performance of MoROA is compared to existing RAW optimization algorithms and traditional 802.11 channel access methods. The results show that the trained surrogate models can accurately predict RAW performance with a relative error less than 7% and 10% for 90% and 95% of the RAW configurations respectively. Compare to the traditional 802.11 channel access functions, by using the objective weight and assigning stations into homogeneous groups, MoROA achieves either 65% higher throughput, or 96% less energy consumption, or a weighted solution in between, in dense heterogeneous networks.

Keywords: IEEE 802.11ah, restricted access window (RAW), surrogate model, real-time station grouping, multi-objective optimization

1. Introduction

The recently released long-range and low-power Wi-Fi standard IEEE 802.11ah proposes a novel channel access method, referred to as Restricted Access Window (RAW). It is a flexible hybrid method, highly suited to provide scalable connectivity to both sparsely and densely deployed low-power devices. RAW is based on station grouping and attempts to reduce contention and collisions

Email addresses: le.tian@uantwerpen.be (Le Tian), michael.mehari@ugent.be (Michael Mehari), Serena.Santi@uantwerpen.be (Serena Santi), Steven.Latre@uantwerpen.be (Steven Latré), eli.depoorter@ugent.be (Eli De Poorter), jeroen.famaey@uantwerpen.be (Jeroen Famaey)

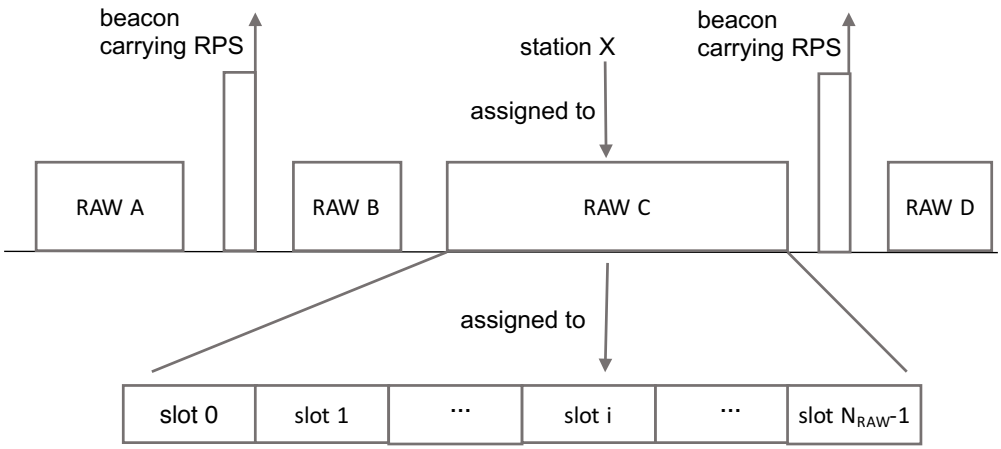


Figure 1: Schematic representation of the RAW mechanism.

in highly dense deployments by dividing stations into groups and allowing channel access to one group at a time. Consequently, IEEE 802.11ah allows up to 8192 stations to connect to a single Access Point (AP).

Figure 1 schematically depicts how RAW works. Specifically, the channel airtime is split into several intervals, some of which are assigned to RAW groups, while others are shared and can be accessed by all stations using the traditional 802.11 Enhanced Distributed Channel Access and Distributed Coordination Function (EDCA/DCF), which rely on carrier-sense multiple access with collision avoidance (CSMA/CA) channel access. At fixed intervals a beacon frame is transmitted, carrying a RAW Parameter Set (RPS) information element. The RPS specifies the stations belonging to each group using the start and end association ID (AID), the group start time, and duration. Moreover, each RAW group consists of one or more equal-duration slots, among which the stations assigned to the RAW group are evenly split (using round robin assignment). The RPS information element also contains the number of slots, slot format and slot duration count sub-fields, which jointly determine the RAW slot duration. Moreover, RAW supports two types of transmissions indicated by the cross slot boundary (CSB) sub-field of the RPS information element. When the CSB sub-field is set to false, stations should not start a transmission if the remaining time in the current RAW slot is not enough to complete it. This remaining time is termed as holding period. Alternatively, stations are allowed to continue their ongoing transmissions even after the end of the current RAW slot when the CSB sub-field is set to true. For a more in-depth description of RAW, the reader is referred to existing literature [1, 2].

The 802.11ah standard, however, does not specify how to configure the actual RAW grouping parameters. Additionally, previous research has shown that the optimal RAW configuration depends on a variety of network-related parameters, such as the number of stations, traffic patterns, and network load [3]. Incorrect configuration severely impacts throughput, latency and energy efficiency. As such, there is a need for RAW optimization algorithms that collect network-related information, and at the start of each beacon interval adapt the RAW configuration based on the current network conditions. Such an algorithm should be able to calculate a new RAW configuration in real-time (i.e., at most a few milliseconds), as it needs to use network-related information obtained from the previous beacon interval and calculate a solution before the new RPS information element is broadcast. Moreover, in order to select the optimal RAW parameters, it should be

able to predict RAW performance for a given set of parameters under specific network and traffic conditions. This is achieved using some sort of model of the environment, which takes as input network conditions and a RAW configuration, and generates as output one or more performance metrics (e.g., throughput or energy consumption).

In the past, several analytic models have been proposed to predict RAW performance [4, 5]. However, such models rely on simplifications and unrealistic assumptions (e.g., no capture effect, no hidden nodes, homogeneous stations, saturated or static traffic). To address these problems, we proposed a novel algorithm for RAW optimization based on surrogate modeling, named Model-Based RAW Optimization Algorithm (MoROA) [6]. A surrogate model is based on supervised learning (e.g., Kriging, or neural networks), but can be accurately trained with very few labeled sample data points [7] . This is important, as a RAW configuration depends on many input variables that can take a wide range of values. As there is no IEEE 802.11ah hardware available, training is done using the realistic IEEE 802.11ah ns-3 model [8]. It implements the capture effect, hidden stations, as well as heterogeneous stations and traffic loads. MoROA uses generic and flexible surrogate models of throughput to determine the optimal RAW configuration, resulting in significant throughput improvements. However, in addition to throughput, which was the focus in this previous work, energy efficiency is also a major concern in IoT networks, as sensors are often battery powered and they should operate for months or even years on a single battery charge. Therefore, in this paper we present an extended version. As a first contribution, it generates surrogate models of both throughput and energy consumption. We specifically improved the training point selection methodology to speed up the training process for the energy model, which suffers from many non-linear regions in its output parameter. As a second contribution, it clusters stations into groups based on their traffic characteristics, and determines the optimal RAW configuration for throughput and energy efficiency based on the trained models by solving a non-linear constrained optimization problem. In contrast to existing algorithms, the extended MoROA supports multi-objective optimization of both throughput and energy, dynamic and heterogeneous traffic. Moreover, MoROA can account for stations with heterogeneous modulation and coding schemes (MCSs) and packet sizes, by clustering stations with similar characteristics into the same RAW group.

The remainder of this paper is structured as follows. Section 2 surveys related work in terms of RAW performance modeling and optimization algorithms and compares them to our contributions. Section 3 details the methodology used to define and train the surrogate models for both throughput and energy consumption, elaborating on the output parameters design for the surrogate models. Section 4 presents the extended MoROA, describing the multi-objective optimization approach for optimizing both throughput and energy performance. Section 5 evaluates the accuracy of our presented models of throughput and energy consumption, comparing it to simulation results, and analyses the impact of different RAW configurations on throughout and energy consumption for deriving the optimal RAW configuration. Moreover, performance of MoROA is evaluated and compared to state of the art RAW optimization algorithms as well as the traditional EDCA/DCF function of IEEE 802.11, in terms of both throughput and energy consumption. Finally, Section 6 offers conclusions and a short overview of future work.

2. Related Work

Since the RAW feature was proposed, several studies have been conducted on the evaluation of RAW performance. Raeesi *et al.* demonstrate that the RAW mechanism can provide substantial

improvements in terms of throughput, delay and energy consumption, in particular in highly-loaded dense network scenarios [9]. In our own previous work [3], we further evaluated the optimal RAW station grouping configuration under a variety of traffic conditions, such as traffic load, number of stations and RAW group duration on the optimal number of RAW groups. These works prove the strong correlation between network and traffic conditions on one hand, and the optimal RAW configuration on the other. This supports the hypothesis that there is a need for real-time RAW parameter optimization.

To determine the optimal RAW parameters, several analytical models have been proposed to calculate RAW performance under specific network and traffic conditions. These models make use of different techniques, such as probability theory [5], Markov chains [4, 10, 11], and maximum likelihood estimation [12]. Early works assume the network is operating under saturated state, where each station always has packets to send [11, 10, 12]. This is not a very realistic assumption for Internet of Things (IoT) and machine-type communication (MTC) [4]. The model proposed by Zheng *et al.* considers both cross and non-cross slot boundary traffic, and is able to calculate the throughput with any given number of stations and RAW duration [11]. A more accurate mathematical model was recently developed by Lyakhov *et al.* [10], by taking into account the non-steady state of the back-off function at the beginning of the RAW period. Park *et al.* determine the RAW group duration for a certain number of stations to get maximized successful transmission probability [12]. In contrast, more recent works assume each station sends one packet per RAW slot interval [4, 5, 13]. Khorov *et al.* built a model to analyze the successful packet transmission probability under a given RAW group duration [4]. The model of Wang *et al.* focuses on energy consumption [5]. Chang *et al.* took a step further, supporting more diverse traffic demands [14]. They use the results of two extreme cases (i.e., infinite traffic and one packet sent per RAW slot) to extrapolate a regression-based analytical model that can accurately fit the contention success probability of any traffic patterns. However, the regression model does not take the finite length of the RAW slot into account.

All existing analytical models share two main shortcomings. First, they are computationally hard. This makes it unfeasible to execute them in real-time on actual AP hardware, where at most a few milliseconds are available at the start of the beacon interval to calculate a new RAW configuration. More importantly, they assume ideal channel conditions, without communication errors, delays or capture effects, or did not takes into account the important peculiarity of the RAW mechanism. The combination of these factors make such models useful only from a theoretical point of view, to analyze the effectiveness of RAW under a variety of conditions. However, they cannot be used for real-time station grouping under dynamic and realistic traffic conditions. Our proposed surrogate modeling approach aims to address both of these issues.

In addition to modeling RAW performance, it is necessary to use this information in real-time, in order to optimize RAW parameters in an actual network. Current solutions are mainly based on set partitioning. These RAW optimization algorithms assume the number of RAW slots and groups is predetermined, and decide how to partition the associated stations among them, according to some metric. Their simplicity makes it computationally feasible to deploy them in real networks. Several algorithms utilize RAW to mitigate hidden node collisions by splitting mutually hidden nodes into orthogonal groups [15, 16]. Chang *et al.* proposed a set partitioning algorithm that assumes the (static) traffic demand of each station is known by the AP and load balances them across groups [17]. Other existing algorithms focus on simple partitioning metrics, such as fully random [18] or based on the back-off timer value [19], which in reality is not known to the AP.

Such set partitioning algorithms have several shortcomings. First, high channel contention exists in dense sensor network even without the presence of hidden nodes. Reducing hidden nodes can mitigate collisions to some extent, but is not sufficient. Second, they expect all information, such as the exact traffic intensity of each station, to be readily available at the AP side, which in reality is not the case. Third, they assume that the number of groups and slots as well as their duration are predefined, and only the partitioning of stations among them needs to be solved. The number of groups and their duration, however, significantly influence RAW optimality [3]. Finally, none of the presented algorithms take into account traffic dynamics. In a real network, the upstream traffic intensity of stations may change over time for a variety of reasons, and the algorithm should therefore adapt to these changes.

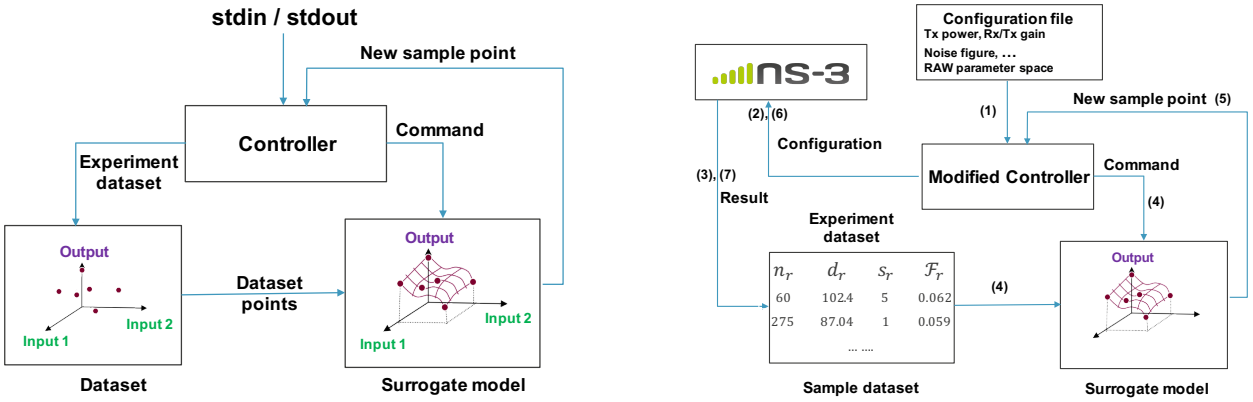
We had proposed the Traffic-Aware RAW Optimization Algorithm (TAROA) [2, 20]. It adapts the optimal RAW parameters in real-time by estimating the current traffic conditions, based solely on information available at the AP. However, it derives the optimal number of stations assigned to a group based on saturated state simulation results on throughput. Recently, we proposed a RAW optimization algorithm based on surrogate modeling [6]. Different from our previous work [2, 20], it uses surrogate models to determine the optimal RAW configuration, in order to maximize the throughput. Moreover, it supports a wide range of traffic conditions and heterogeneous MCS and packet sizes. However, energy efficiency is also a major concern in IoT networks. Therefore, in this article we present an extended version. It generates surrogate models of both throughput and energy consumption, and determines the optimal RAW configuration by optimizing both throughput and energy simultaneously.

3. Surrogate Model of RAW performance

This sections introduces the surrogate modeling approach and toolbox, as well as its integration with the ns-3 network simulator. Subsequently, we describe how surrogate modeling can be used to train a performance model using supervised machine learning, for estimating throughput and energy consumption of the IEEE 802.11ah RAW under a wide range of network and traffic conditions.

3.1. Surrogate model training methodology

In order for the AP to determine the optimal RAW parameters in real-time, a RAW performance model is needed. Given the current network conditions (e.g., network topology, traffic) and a set of RAW parameter values (e.g., number of groups and slots, group duration, station assignment) the model should estimate performance (e.g., in terms of throughput and energy). Based on this model, optimal RAW parameters can subsequently be derived for different objectives based on current network conditions. However, existing analytical models are computationally expensive and unrealistic due to their assumptions. Surrogate modeling provides the answer [7]. A surrogate model is trained at design time, using a limited number of input-output sample data points obtained through simulation or real-life experiments. Surrogate modeling is especially suited for tasks with a large input space, as an accurate model can be trained based on relatively little input data points. Moreover, evaluating the model at runtime is computationally efficient, equivalent to a constant-time table lookup. This makes surrogate modeling highly suitable for RAW performance modeling, as the input space is very large, and efficient runtime model evaluation is needed for real-time RAW parameter selection. Additionally, by using realistic simulation results, a surrogate model does not suffer from the same restrictive assumptions as existing analytical models.



(a) General SUMO modeling architecture.

(b) Integrated RAW model training using SUMO and ns-3.

Figure 2: Training methodology used to model RAW performance.

The Matlab Surrogate Modeling (SUMO) Toolbox is a flexible framework for accurate global surrogate modeling [7]. The general SUMO modeling architecture is illustrated in Figure 2a. The controller plays the key role, managing the modeling process. First, the user offers a set of initial sample data points (including input and output) to the controller. The controller uses those points to construct an initial surrogate model. Next, with the constructed model, the controller predicts the next input space element from which the expected accuracy improvement is the largest. The modeling process keeps iterating, and terminates once certain stopping conditions are met (e.g., the maximum training time is exceeded).

To train the RAW model, we used our previously developed IEEE 802.11ah ns-3 simulation module [8]. Figure 2b shows our adapted training methodology to allow the integration between ns-3 and the SUMO toolbox. The modified controller conducts similar tasks as the original SUMO controller. However, it now directly interfaces with ns-3. When a new sample data point is generated for which the output is unknown, the controller will initiate an ns-3 simulation to determine the output associated with the input parameter values of the data point. The SUMO toolbox executes the following steps to train the RAW performance surrogate model, using the same numbering as the arrows in Figure 2b:

1. The controller reads the settings of the 802.11ah RAW experiment, including the general parameters of 802.11ah (cf. Table 1) and the input space parameters (cf. Table 2).
2. The controller generates an initial sample set based on the input space, and starts ns-3 experiments with the required settings.
3. At the end of each experiment, the controller retrieves the evaluation criterion (e.g., throughput, energy consumption) of the experiment and builds the sample data space.
4. After the experiments with the initial sample data set, the controller builds the surrogate model and calculates the cross validation score.
5. The sampling strategy is applied to select the next 10 sample data points to improve the model accuracy. The step size of 10 is selected based on trail and error.
6. The controller starts the next ns-3 experiments for the newly selected sample data points.
7. The controller reads the output of the experiment and updates the sample data space, then goes back to step 4. This process continues until the stop conditions are met.

In our experiments, the SUMO toolbox is configured to use the latin hypercube sampling method [21] to generate 100 initial sample data points, Kriging interpolation is used to train the model [22], FLOLA-Voronoi sampling for generating the next sample points [23], and the 10-fold cross-validation with a root-relative square error (RRSE) measure to evaluate the model accuracy [24]. The training stops once the cross-validation score remains stable (2 digits of precision) for 10 successive iterations (i.e., 100 sample data points), or the number of training data points exceeds 2500.

3.2. Training scenario

In this section, we provide an overview of the static simulation environment parameters used during training. Since the goal of RAW is scalability under uplink traffic, we consider an IoT sensing scenario, where sensors periodically monitor the environment and send the resulting data to a server (via the AP). The PHY and MAC layer parameters are shown in Table 1. Given the low-power nature of battery powered sensors, the PHY layer parameters are configured based on the low-power 802.11ah radio hardware prototype developed by Ba et al. [25], with a transmission power of 0 dBm, a gain of 0 dBi (for both sensor and AP), and noise figure of 6.8 dB. In order to obtain a model that is independent of the actual deployment of stations, stations are randomly placed around the AP within a maximum radius. The size of the stations' transmit queues is configured to be 10 packets. As the RAW duration can be shorter than the beacon interval, there are still some remaining shared airtime outside of the RAW group. Allowing stations to transmit during this RAW-free period would alter performance results. As our goal is to purely model the performance of the RAW period itself, we disable packet transmissions during the shared airtime during training. Since no off-the-shelf IEEE 802.11ah hardware is available, we use energy consumption values for each radio state (i.e., transmitting, receiving, idle and sleeping) from the experimental IEEE 802.11ah transceiver developed by Ba et al. [26].

As the RAW optimization algorithm proposed in Section 4 groups together stations that use the same MCS, the model assumes a fixed MCS. However, as RAW performance depends on the MCS used, a different model is trained for each MCS that stations are expected to use. We illustrate this by developing a separate high-throughput (HT) and low-throughput (LT) model for two different MCS parameter sets. This can be trivially extended to other MCS values. For training simplicity, we assume each station sends one packet per second. However, we show in Section 4 how this model can be used to calculate RAW performance under arbitrary data transmission intervals.

Each experiment runs for 60 seconds of simulated time. As RAW is configured in each beacon interval of 204.8 ms, the results of every simulated configuration are averaged over 290 beacon intervals, ensuring the generality of the trained model.

3.3. Input and output parameters for RAW modeling

The surrogate model aims to accurately predict performance t_r (e.g., throughput, energy consumption) for a RAW group r with duration d_r, consisting of s_r slots, and with n_r stations assigned to it. The resulting model can be represented as functions $\mathcal{F}$, as follows:

$$t_r = \mathcal{F}(n_r, d_r, s_r) \tag{1}$$

In addition to throughput and energy consumption, the simulator calculates a variety of other performance metrics, such as packet loss and latency. As such, the same methodology as described here can be used to train a model for predicting these other performance metrics. However, to simplify the explanation, we focus on throughput and energy.

Table 1: Simulation parameters used during training

PHY parameters	**Value**
TX power (dBm)	0
TX/RX gain (dB)	0
Noise Figure (dB)	6.8
Propagation model	Outdoor
Error rate model	YansErrorRate
MAC parameters	**Value**
Traffic access categories	AC_BE
Duration of AIFS (μs)	264
Duration of SIFS (μs)	160
Beacon interval (ms)	204.8
Size of transmission queue (packets)	10
Packet transmission interval (s)	1
Station distribution	random
Cross-slot boundary	false
Low-Throughput (LT) parameters	**value**
Wi-Fi mode	MCS1, 1 Mhz
Average payload size (bytes)	64
Topology radius (m)	200
High-Throughput HT parameters	**value**
Wi-Fi mode	MCS9, 1 Mhz
Average payload size (bytes)	256
Topology radius (m)	80

3.3.1. Input parameters design

To build the SUMO model, the input parameter space needs to be defined. It consists of the minimum and maximum value of each parameter, as well as a step size. The minimum and maximum can be determined based on expert knowledge of RAW performance, as well as legal values defined by the IEEE 802.11ah standard. The range of the number of stations n_r in a RAW group should span from low to high traffic conditions, so the trained model gives accurate results independent of the density. From our previous studies on RAW performance [3], and based on the parameters listed in Table 1, a minimum value of 60 and maximum value of 400 stations per group were deemed to cover all possible traffic conditions. The RAW group duration should be large enough to send at least 1 packet successfully, and at most equal to the duration of the beacon intervals. As such, d_r is varied between 40960 μs and 204800 μs. The number of slots s_r is bound between 1 and 64, as per the IEEE standard. However, a very high number of slots leaves not enough time within a slot to successfully transmit a packet. As such, we limit s_r between 1 and 50.

The actual step size of n_r and s_r is 1, as they are integer variables. The parameter d_r has a step size of 120 μs, as defined in the 802.11ah standard. This results in a total input space of 2.3×10^7 possible data points. This is too high to properly train the model in a feasible amount of time. To alleviate this, we experimentally determined a good step size for each of the three parameters, leveraging the trade-off between accuracy and training speed. For n_r and s_r a step size of 5 was selected. It was found that the RAW duration d_r has a high sensitivity. As such, a small value of 5120 μs was chosen as its step size. This results in a significantly reduced input space of 25047 data points. Table 2 summarizes the selected input parameter space. Note that a slot count s_r equal to 0 is not legal, and s_r therefore takes on values from the set $\{1, 5, 10, ..., 50\}$. Results for

Table 2: Definition of the input parameter space

Parameter	Min	Step	Max
n_r	60	5	400
d_r (μs)	40960	5120	204800
s_r	1	5	50

data points outside the reduced input parameter space are obtained via linear interpolation of the two nearest data points included in the model.

3.3.2. Output parameters design

As non-linear regions are more difficult to model compared to linear regions, more sample points from the non-linear regions are required during the training process. Therefore, a good selection of output parameters, which leads to more linear regions, can speed up the training process. For our models, there are several options for the output parameters, which can directly or indirectly represent the performance metrics (i.e., throughput and energy) that we focus on. As in our modeling process, RAW group duration may be shorter than the beacon interval, we can use throughput over the RAW groups or throughput over the whole beacon interval as output parameter for the throughput model. The latter one results in more linear output than the first one, since RAW group duration varies from data points to data points, while the beacon interval is constant. Similarly, energy consumption per packet or total energy consumption can be considered as output parameter for the energy model. While the former one is the ratio between the latter one and the number of successfully transmitted packets, leading to more non-linear output. Therefore, we choose throughput over the whole beacon interval and total energy consumption as output parameters for the throughput and energy model respectively.

4. Model-Based RAW Optimization Algorithm

This section introduces the Model-Based RAW Optimization Algorithm (MoROA). It relies on the same principles as our previously proposed RAW optimization solutions TAROA [2] and Enhanced Traffic-Aware RAW Optimization Algorithm (E-TAROA) [20]. As such, it is also traffic-aware and able to adapt to changing traffic conditions. Moreover, MoROA supports heterogeneous stations with variable MCS and packet size. Finally, the extended algorithm presented here differentiates from our original version of MoROA [6] in the fact that it finds the optimal RAW configuration by optimizing both throughput and energy simultaneously, instead of only focusing on a single objective.

4.1. Overview

As in Section 3, we assume an IoT sensor-based monitoring scenario. However, in contrast to the model presented above, the algorithm is able to handle heterogeneous stations, with variable transmission intervals, MCS, and packet sizes. This is achieved, on one hand, by combining different trained models for different types of stations, and on the other hand, by transforming the performance metric output based on traffic conditions. Moreover, the data transmission interval of sensor stations can change over time (e.g., when an environmental event triggers a change in the sensor measurement interval). The goal of RAW optimization is to assign stations to a set of RAW groups with appropriate RAW parameter configurations, in order to achieve the required objective (e.g., maximum throughput, fairness, or minimum energy consumption).

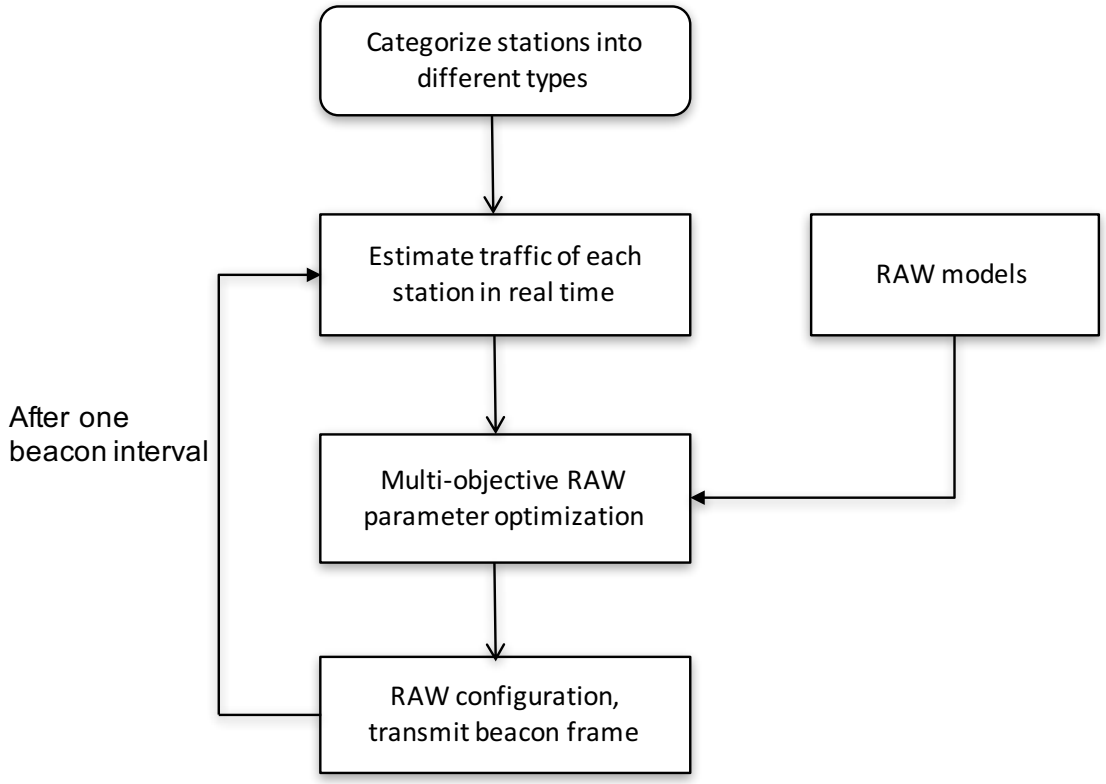


Figure 3: Overview of the steps involved in MoROA.

The proposed algorithm uses only information readily available at the AP. The concrete steps are illustrated in Figure 3. The algorithm is executed at each target beacon transmission time (TBTT) and supports dynamic network conditions (i.e., stations joining or leaving the network) and dynamic traffic (i.e., stations that change their transmission interval). First, the AP categorizes stations into different groups. In MoROA, this is based on MCS and packet size, as these two factors influence the minimum time needed to successfully transmit a packet, and therefore the optimal slot duration. However, other grouping strategies can be used as well. Second, the AP determines the traffic of each station, and selects the stations that are expected to have pending packets to transmit in the next beacon interval. As this information is not readily available to the AP, it has to be estimated. We apply a traffic estimation method for IoT sensor traffic proposed in our previous work [20]. Finally, we utilize RAW performance models $\mathcal{F}(n, d, s)$ that take as input the number of stations n, the group duration d, and the number of slots s, and gives as output some performance metric t (e.g., throughput, energy). This function serves as the basis for an optimization problem, that optimizes the RAW parameter decision variables n, d and s, in order to maximize the selected performance metrics. The optimization is subjected to a set of constraints that in turn ensure the traffic estimation for the following beacon intervals is accurate. The output of the algorithm is a RAW configuration consisting of a set of groups, containing for each group a set of assigned stations, group duration, and the number of slots. Note that in the remainder of the description, we assume the use of the surrogate model described in Section 3. However, any function $\mathcal{F}(n, d, s)$ that satisfies the requirements on input and output parameters and that can be calculated in real-time could be used in combination with MoROA.

4.2. Multi-objective RAW parameter optimization with heterogeneous stations

We assume a set of stations $\mathcal{N}$ associated with the AP. At the start of each beacon interval b, stations are split into k distinct clusters, with each cluster $i \in [1, k]$ consisting of the stations $\mathcal{N}_i \subseteq \mathcal{N}$. This can be achieved using any clustering algorithm, based on a variety of distance metrics. We use standard K-means clustering combined with the packet transmission time as a distance metric. The packet transmission time can be trivially calculated based on MCS and packet size, both of which can be monitored at the AP. This results in stations with the same MCS and average packet size to be clustered together, which is an assumption of our current surrogate model. In future work, we plan to train models without this assumption, allowing a wider variety of clustering approaches.

Subsequently, by taking the fairness into account, the AP determines which stations $\mathcal{N}_i^b \subseteq \mathcal{N}_i$ of each cluster i are predicted to have packets queued for transmission during the next beacon interval b. This can be done using our previously proposed traffic estimation method for IEEE 802.11ah [20]. We also define $n_i^b = \left|\mathcal{N}_i^b\right|$ as the number of stations in cluster i predicted to have packets queued for transmission during the next beacon interval b. Finally, the algorithm assigns a RAW group to each cluster i. It calculates the number of stations n_i that will be allowed to access the channel, the duration d_i of the group, and in how many slots s_i to split the group.

Finding the RAW parameter values that optimize the chosen performance metrics (i.e., maximizing throughput and minimizing energy) can be formulated as a multi-objective optimization problem. As the sum of all group durations should not be higher than the beacon interval duration d_b, the chosen duration d_i of a cluster i influences the RAW duration of other clusters. A common approach to solving multi-objective optimization problems is to first generate a Pareto front set from which the most desirable trade-off point is selected. A Pareto front set is a set of solutions where any improvement in one objective results in the worsening of at least one other objective. There have been many methods suggested for selecting the optimal Pareto front point, such as the weighted global criterion method, weighted sum method, lexicographic method and weighted min-max method [27]. These methods allow the user to specify preferences, which may be articulated in terms of goals or the relative importance of different objectives. We define an objective function that maximizes throughput and minimizes energy consumption for each group, using a hybrid method combining weighted sum and min-max. However, MoROA could be trivially extended to use a different optimization objective functions, such as for example airtime fairness.The problem can now be formulated as follows:

$$\max\left(\alpha \times \mathcal{Q}_t - (1-\alpha) \times \mathcal{Q}_e\right) \tag{2}$$

With:

$$\mathcal{Q}_t = \min_{i \in [1,k]} \left(\frac{\mathcal{F}_t\left(n_i, d_i, s_i\right)}{n_i^b \times l_i} \right) \tag{3}$$

And

$$\mathcal{Q}_e = \max_{i \in [1,k]} \left(\frac{\mathcal{F}_e(n_i, d_i, s_i) \times l_i}{\mathcal{F}_t\left(n_i, d_i, s_i\right) \times T_c} \right) \tag{4}$$

Subject to:

$$\forall i \in [1, k] : n_i \leq n_i^b, \quad 1 \leq k \leq 85 \tag{5}$$

$$10 \leq \sum_{i=1}^{k} n_i \leq 100 \tag{6}$$

$$\sum_{i=1}^{k} d_i \leq d_b \tag{7}$$

$$\begin{cases} d_i \leq 1969.12, & if \quad 1 \leq s_i \leq 8 \\ d_i \leq 1990.4, & if \quad 9 \leq s_i \leq 64 \end{cases} \tag{8}$$

$$\forall i \in [1, k] : p_i^s < \frac{n_i \times l_i}{\mathcal{F}_t\left(n_i, d_i, s_i\right)} \tag{9}$$

Where $\mathcal{F}_t\left(\cdot\right)$ and $\mathcal{F}_e\left(\cdot\right)$ represent the RAW model functions that predict throughput and energy consumption. The variable l_i is the average packet size of stations in cluster i. The continuous variable $\alpha \in [0, 1]$ is a weight used to define the relative importance of both sub-objectives. The parameter d_b represents the duration of the beacon interval b. In Eq. 9, p_i^s represents the successful packet transmission probability of RAW group i. This constraint is required, as the traffic estimation method we use, does not work properly under high packet loss due to contention [2]. As such, we use $p_i^s = 0.99$. When using other traffic estimation methods, this constraint may not be needed.

Q_t represents the objective of throughput, while Q_e represents the objective of energy consumption per packet. Both objectives try to find a balance among different groups. The objective function, as illustrated in Eq. 2, seeks to maximize Q_t and minimize Q_e. T_c denotes the training time. RAW groups are allowed to use all the airtime (i.e., the entire beacon interval), as represented by Eq. 7. A RAW-free period can be trivially introduced by changing this constraint. Eq. 5 indicates that the number of stations that are allowed to access the channel should not exceed the the number of stations predicted to have packets queued for transmission. Note that this formulation assumes that stations will only attempt to transmit one packet per beacon interval. This assumption generally holds for sensor scenarios, where the throughput of individual stations is low [4]. In reality, some stations may have multiple packets queued, especially when the traffic estimator is still learning [20]. However, this has a negligible effect on performance of the algorithm over longer periods. Currently, MoROA does not consider the required sequentially of AIDs in the RPS element, as we consider AID reassignment a separate issue left for future work.

The formulated problem is a non-linear constrained optimization problem with integer decision variables (i.e., n_i, d_i, and s_i). This can for example be solved using genetic algorithms. A relaxed version of the problem with continuous decision variables could alternatively be solved using for example the Interior-Point method, in combination with a rounding strategy to convert the resulting continuous decision variable values to integers. Moreover, by taking the characteristic 802.11ah into account, the value of n_i and k can be further limited, as depicted in Eq. 5, 6 and 8 . It reduces the number of potential solutions and therefore the solving time. The sum of n_i for all the RAW groups is limited to the maximum number of packet n_{max}^b that can be successfully transmitted during one beacon interval. Due to the relatively low data rates supported by 802.11ah (i.e., up to 7.8 Mbps for a 2 MHz bandwidth), n_{max}^b ranges from around 10 to 100 in practice. For homogeneous networks, we can use the corresponding n_{max}^b based on the used MCS. For heterogeneous networks in which multiple MCSs are used, we can use the n_{max}^b (equal to 100) of MCS 8 for a

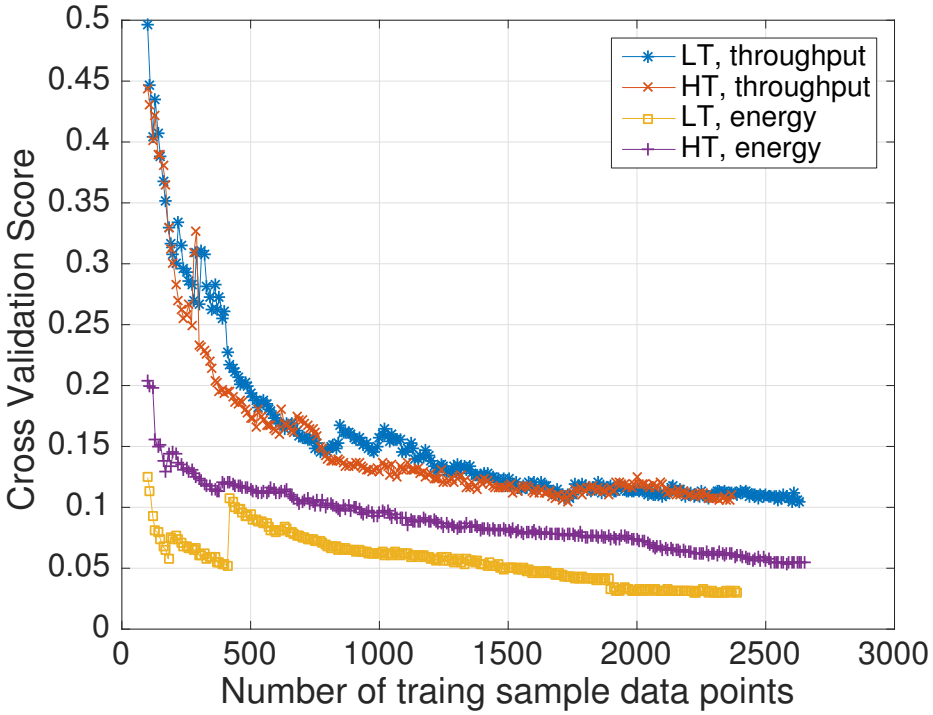


Figure 4: Cross validation score of the surrogate models as a function of the number of training sample data points.

2 MHz bandwidth. In the RPS element, each RAW group definition requires between 3 and 12 bytes. As the maximum size of the RPS elements is 256 bytes, the number of groups k cannot be higher than 85. Based on the specification of RAW, either the parameter s_i is at most 64 and d_i is at most 1990.4 ms, or the parameter s_i is at most 8 and d_i is at most 1969.12 ms. While the minimal value of d_i is 0.62 ms.

5. Performance Evaluation and Discussion

This section presents the evaluation results of the RAW performance of surrogate model and the MoROA RAW optimization algorithm. First, the simulation setup is discussed. Second, the accuracy of the surrogate model is compared to simulation results. Finally, MoROA is evaluated and compared to state of the art RAW algorithms, as well as the traditional EDCA/DCF method of IEEE 802.11.

5.1. Simulation setup

All evaluations are performed using our previously developed IEEE 802.11ah ns-3 module [8], based on ns-3 version 3.23. We consider the same IoT scenario as described in Section 4. The same default PHY and MAC layer parameters used as shown in Table 1. We consider both homogeneous and heterogeneous scenarios. Homogeneous scenarios are used to validate the surrogate model and compare to E-TAROA [20]. In heterogeneous scenarios, half of the stations use the high-throughput (HT) settings and half of them use the low-throughput settings (LT) listed in Table 1. The data transmission interval of each station is selected uniformly at random from the interval $[1, 10]$ seconds.

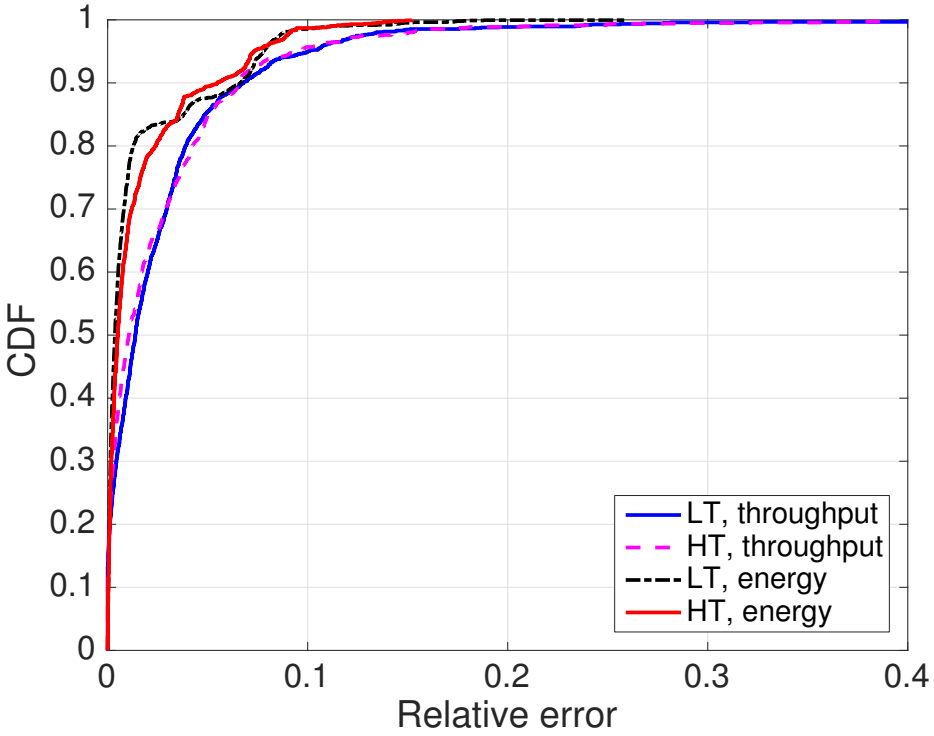


Figure 5: Performance comparison between the surrogate model and simulation results for 5000 random test data points.

RAW performance is evaluated in terms of two metrics: throughput and energy. Throughput is calculated as the average number of successfully received application payload bits by the AP per second. Energy is considered as the total energy consumption averaged over the number of successfully received packets. Each simulation runs for 600 seconds, and all results are averaged over 10 iterations, with the variability of results over these iterations quantified using the standard deviation (SD).

5.2. SUMO model validation

In this section we evaluate the training convergence of the surrogate model, as well as its accuracy compared to simulation results. Figure 4 plots the model's cross validation score as a function of the number of training samples used. The cross validation score provides a measure for the accuracy of the resulting model on predication of RAW performance using the IEEE 802.11ah ns-3 simulator. A consistently low score signifies that the training process has converged. The results show, for the throughput models of both HT and LT stations, that the convergence occurs after around 1700 training samples have been used. This comes down to about 0.0074% of all data points in the input space (i.e., 2.3×10^7), and about 6.8% of the reduced data space (i.e., 25047) from which samples were drawn during training. The convergence of energy models happens after 1900 and 2500 training samples have been used for energy models of HT and LT stations, respectively. The training of the HT throughput model and LT energy model stopped after 2370 and 2400 iterations training samples respectively, as it satisfied the cross-validation stop conditions, i.e., 10 consecutive cross-validation scores (2 digits of precision) remain the same. The training of the other models stopped after the maximum number of 2500 iterations training samples , having

achieved 15 consecutive cross-validation scores between 0.11 and 0.12 for the LT throughput model, and 9 consecutive cross-validation scores between 0.054 and 0.058 for the HT energy model. The cross-validation scores decrease as more training samples are used, is due to the sampling strategy FLOLA-Voronoi. It selects points from non-linear regions and the sparsely sampled regions, making the model more and more accurate. It should be noted that the cross-validation scores fluctuate sometimes. This is because new training sample points are selected from an untouched region, and the surrogate model is recalculated to consider this change.

In order to ensure no over-fitting occurred, we evaluated if the surrogate model also provides accurate results for data points outside the reduced input space (cf. Table 2). Figure 5 plots the cumulative distribution function (CDF) of the relative error (i.e., ratio between absolute error and simulation results). In total, 5000 random data points were generated from all 2.3×10^7 possible points. It shows around 95% and 98% of the test data points have a relative error less than 10% for the throughput and energy models for LT stations, respectively. Similar results are obtained for the HT stations. 90% of the test data points have a relative error less than 7% for both the throughput and energy models of LT stations. For the throughput and energy models of HT stations, this is respectively 7% and 5%. These results validate the ability of the surrogate model to estimate RAW performance accurately, even for points outside of the reduced input data space used for initial training.

5.3. RAW configuration analysis

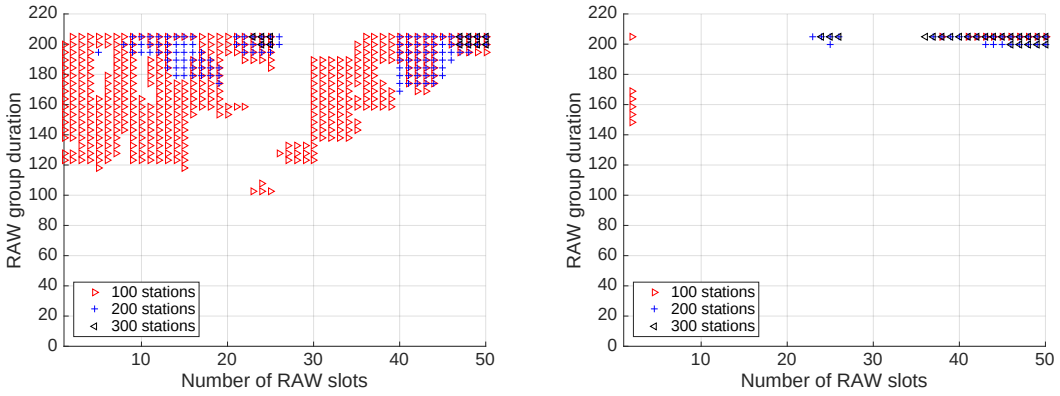
In this section, based on the built model, we analyze the impact of different RAW configurations on throughout and energy consumption, and derive the optimal RAW configuration. For each number of stations, we first find the highest output t_{max} from the built model $\mathcal{F}_t(\cdot)$, and find e_{min} from the model $\mathcal{F}_e^p(\cdot)$, which is calculated as follows:

$$\mathcal{F}_e^p(n_r, d_r, s_r) = \frac{\mathcal{F}_e(n_r, d_r, s_r) \times l_r}{\mathcal{F}_t(n_r, d_r, s_r) \times T_c} \tag{10}$$

The model $\mathcal{F}_e^p(n_i, d_i, s_i)$ predicts the energy consumption per packet for a RAW group r with duration d_r, consisting of s_r slots, and with n_r stations assigned to it. Subsequently, we select the RAW configurations that deviate at most 10% from the optimal throughput t_{max} and the optimal energy consumption e_{min}.

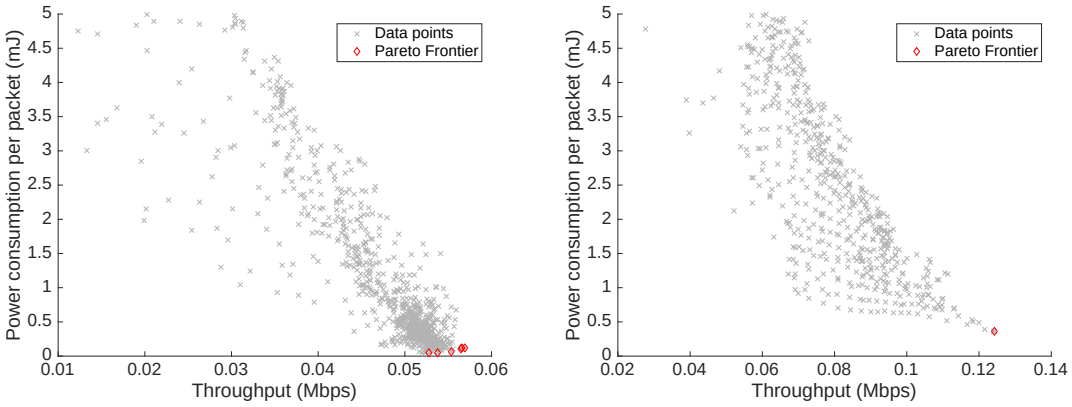
Figure 6 depicts such RAW configurations for 100, 200 and 300 LT stations, respectively. As the figures show, if the number of stations is relatively small (i.e., 100), many different RAW configurations lead to near optimal throughput. This is because channel contention is less fierce in such a scenario, and the RAW configuration thus has less effect on throughput. Even without RAW (i.e., one RAW slot), quite high performance can already be obtained, while a larger number of RAW slots is definitely needed when the number of contending stations increases to 200 or 300). It should be noted that the near optimal number of RAW slots for throughput is discontinuously distributed. For example, for RAW duration 204.8 ms and 200 stations, a number of slots in the intervals 9–16, 21–26 and 40–50 achieve the highest throughput. This is because cross-slot boundary transmissions were disabled, which leads to varying amounts of wasted airtime (i.e., holding time) [10]. Different from throughput, there are fewer near optimal RAW configurations for energy.

There exist some RAW configurations that can maximize throughput and minimize energy simultaneously, as depicted in Figure 7. This figure illustrates the Pareto front of throughput and



(a) Throughput (b) Energy consumption per packet

Figure 6: RAW configurations that deviate at most 10% from the optimal throughput or energy consumption.



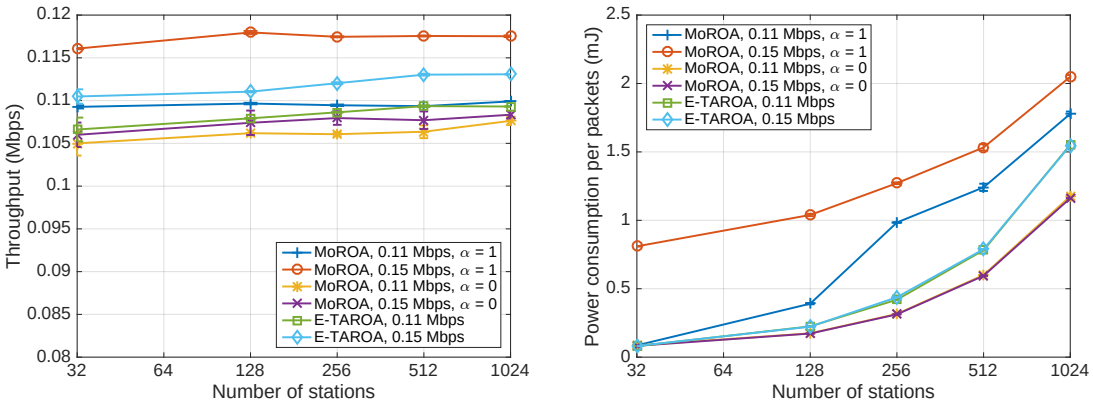
(a) 100 stations (b) 400 stations

Figure 7: Pareto front of throughput and energy per station, pareto front points are highlighted in red.

energy consumption per packet for 100 and 400 LT stations. As shown, the 400 stations have only one Pareto front solution, i.e., the objective of maximizing throughput and minimizing energy consumption can be achieved simultaneously. While for 100 stations there are several Pareto front solutions. Among them, a solution can be chosen based on the relative importance of each of the two objectives, using a multi-objective RAW optimization algorithm such as MoROA.

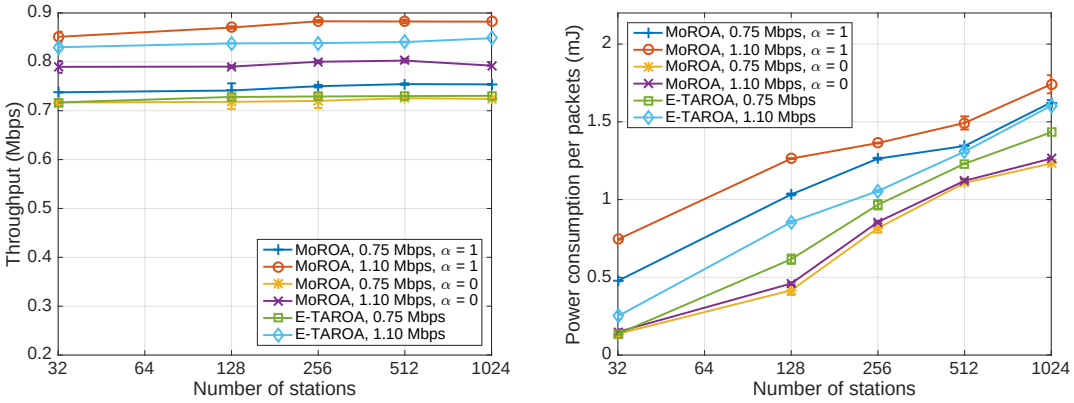
5.4. Homogeneous stations

In this section, we evaluate the performance of MoROA for homogeneous stations (in terms of MCS and packet size) with the multi-objective function weight α (cf. Eq. 2) equals 0 (i.e., pure energy optimization) and 1 (i.e., pure throughput optimization) for a variety of traffic loads and station counts. Two different total traffic loads are simulated, i.e., $T = \{0.11, 0.15\}$ Mbps for the LT scenario, and $T = \{0.75, 1.10\}$ Mbps for the HT scenario. Given the packet payload size and MCS, the maximum throughput that can be achieved is about 0.124 Mbps and 0.90 Mbps



(a) LT, Throughput

(b) LT, Energy consumption



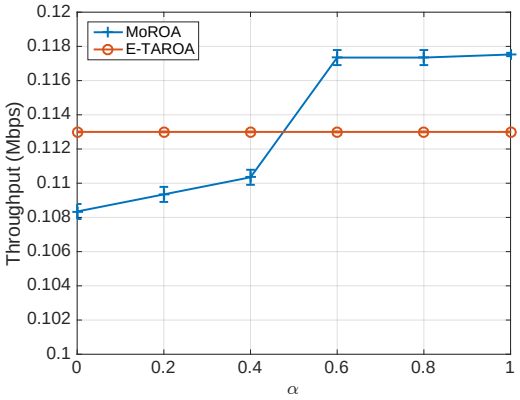
(c) HT, Throughput

(d) HT, Energy consumption

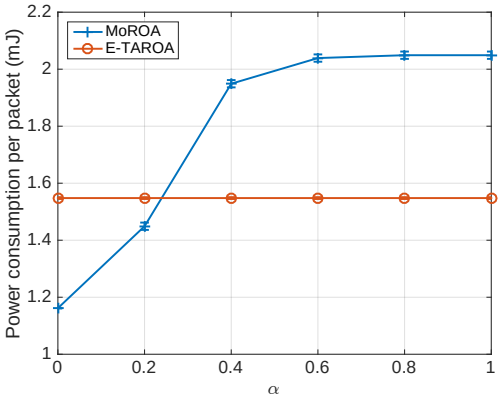
Figure 8: Performance comparison between MoROA and E-TAROA using $\alpha = 0, 1$ for the LT and HT scenarios with different traffic loads and station counts

for LT and HT stations respectively. As such, $T = 0.15$ Mbps and $T = 1.10$ Mbps represent a saturated state, $T = 0.11$ Mbps and $T = 0.75$ Mbps represents a medium to high traffic load. Each station is assigned a random transmission interval, with the ratio between maximum and minimum transmission interval at most 20. As E-TAROA was already shown to outperform EDCA/DCF and other RAW optimization algorithms for homogeneous scenarios, we use E-TAROA as a benchmark in this section [20].

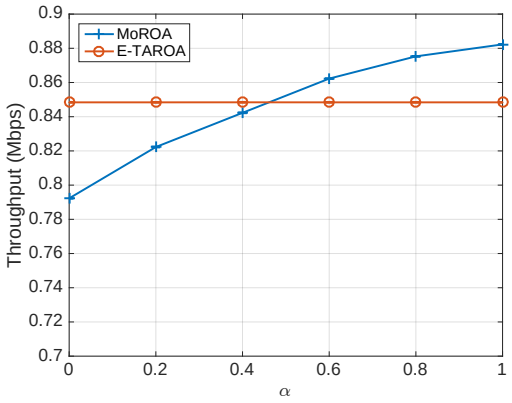
Figure 8 depicts performance in terms of throughput and energy consumption of MoROA and E-TAROA. It clearly shows that, compared to E-TAROA, MoROA can tune the values of α to optimize the required performance objective. For $\alpha = 1$, MoROA improves the throughput up to 5.3% under saturated traffic conditions for any number of stations, for both LT and HT stations. Under medium traffic conditions, MoROA also improves performance. While MoROA consumes more energy than E-TAROA does for both saturated and medium traffic conditions, up to 30%. Conversely, when α is 0, MoROA consumes up to 22.6% less energy at the cost of 4.4% throughput decrease for LT stations, and up to 32.4% less energy at the cost of 3.26% throughput decrease for HT stations. This is due to the fact that E-TAROA derives its optimal RAW parameters from



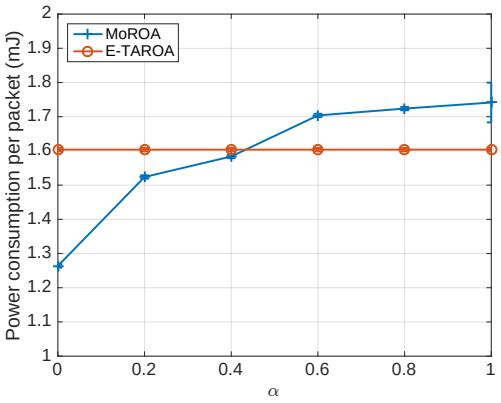
(a) LT, Throughput



(b) LT, Energy consumption



(c) HT, Throughput



(d) LT, Energy consumption

Figure 9: Performance of MoROA as a function of different α

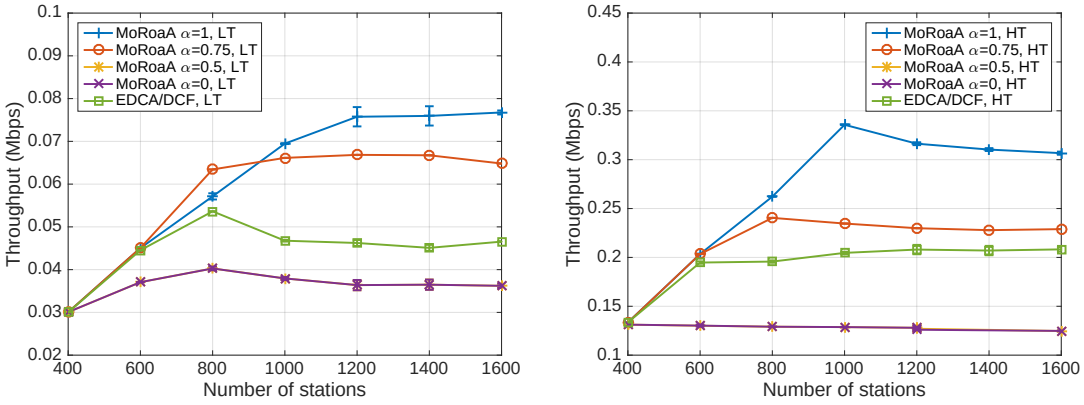
a model of throughput under saturated state as an approximation, rather than optimizing them for different traffic conditions and objectives as the surrogate model does. This results in better overall performance in terms of throughput and energy efficiency. Figure 9 zooms into MoROA performance with different α for 1024 LT and 1024 HT stations, with traffic load 0.15 Mpbs and 1.10 Mbps respectively. It clearly shows, Compared to E-TAROA, that by adjusting the value of α, MoROA can either obtain higher throughput or lower power consumption. Moreover, the energy consumption is significantly affected by α, up to 22.6% and 21.2% can be saved for LT and HT stations respectively . While α has a small impact on throughput, only 4.5% and 4.0% throughput can be improved for LT and HT stations respectively. The above results reveal that, by using surrogate modeling, stations can be assigned to RAW groups in a more optimal way for the desired performance objective by adapting the α value. Moreover, these results show that by employing multi-objective optimization, a small reduction in throughput can lead to a significant improvement in terms of energy efficiency.

5.5. Heterogeneous stations

In this section, we study the performance of MoROA in a network with heterogeneous stations. Moreover, we evaluate the effects of the α weight parameter on throughput and energy consumption of both LT and HT stations, for a variety of network densities. As E-TAROA only supports stations with same MCS and packet size, the results of MoROA are compared to EDCA/DCF in this case.

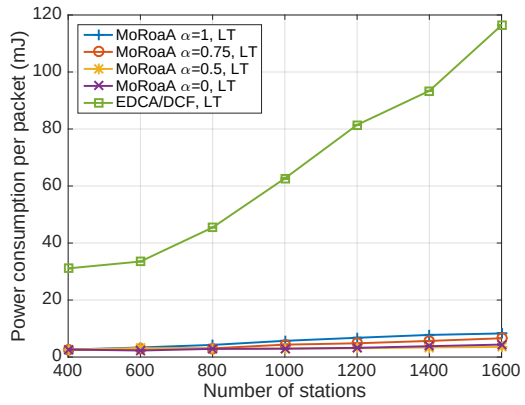
The comparison between MoROA and EDCA/DCF is depicted in Figure 10. With a small number of stations (i.e., 400 stations), MoROA and EDCA/DCF have the same throughput. For 600 stations, EDCA/DCF still remains similar throughput as MoROA with α larger than 0.5. While with more than 1000 stations, the throughput of EDCA/DCF stagnates to about 0.047 Mbps and 0.21 Mbps for LT and HT stations respectively, as the network becomes saturated. However, for $\alpha = 1$ and $\alpha = 0.75$, MoROA reduces the number of collisions and results in better scaling behavior and a higher saturation limit. For example, with 1600 stations and $\alpha = 0.75$, throughput of MoROA is about 38% and 20% higher for LT and HT stations, respectively. If more focus is put on the throughput objective, with $\alpha = 1$, MoROA achieves about 65% and 47% higher throughput for LT and HT stations, respectively. In contrast, when $\alpha = 0$, MoROA obtains 31% and 64% less throughout for LT and HT stations than EDCA/DCF does. In terms of energy consumption, MoROA always significantly outperforms EDCA/DCF. For 1600 stations with $\alpha = 0$, MoROA saves about 96% and 91% energy for LT and HT stations, respectively.

The number of packets that collide with each other were counted during simulation, the results are depicted in Figure 11. It clearly shows that, compare to EDCA/DCF, MoROA significantly reduce the number of collisions. For 400 and 600 stations, the number of collision is 52845 and 60344 for EDCA/DCF. As the number of station increases, the number of collision goes up dramatically, around 1122775 for 1600 stations. In contrast, the number of collision MoROA is at most 29142, 97.4% less than EDCA/DCF. As collision may cause the packets to be dropped, MoROA obtains higher performance (e.g., throughput, energy) than EDCA/DCF. It should be noted that MoROA with $alpha = 0$ have less collision than EDCA/DCF, but lower throughput than EDCA/DCF. This is due to the fact MoROA with $alpha = 0$ tries to avoid the collision as much as possible, in order to minimize the energy consumption. As such, the airtime is not fully utilized, leading to lower throughput.

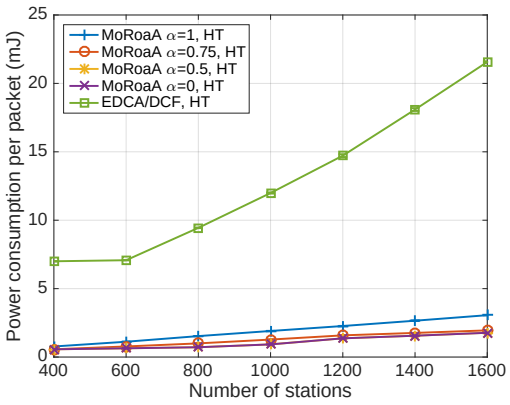


(a) Throughput, LT

(b) Throughput, HT



(c) Energy consumption, LT



(d) Energy consumption, HT

Figure 10: Performance comparison between MoROA using different α and EDCA/DCF, for a heterogeneous network

6. Conclusion and Future Work

In this paper, we present a novel solution for real-time RAW parameter optimization for IEEE 802.11ah, consisting of two contributions. First, we present new RAW performance models for both throughput and energy efficiency, based on supervised surrogate modeling. It can be easily trained on a limited set of labeled data samples. Moreover, it is very fast to evaluate once trained, allowing it to be used for real-time RAW parameter optimization. As a second contribution, we extended our original version of MoROA. It uses surrogate models for the throughput and energy performance metrics to determine the optimal RAW configuration through multi-objective optimization under a variety of network and traffic conditions. Moreover, it supports heterogeneous stations with different MCS and average packet sizes.

The simulation results reveal four key points. First, the built surrogate model for RAW gets high accuracy relative to realistic simulation results. With a training set of 0.0074% of all possible data points, a relative error less than 7% for 90% of the randomly tested RAW configurations is achieved. Second, the built surrogate models reveal that there are many RAW configurations

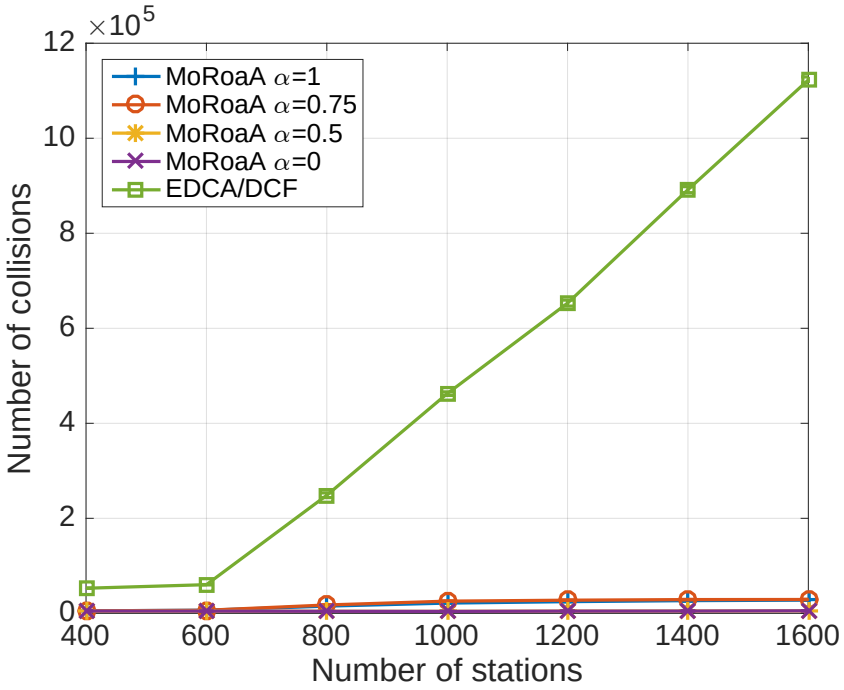


Figure 11: Collision comparison between MoROA using different α and EDCA/DCF, for a heterogeneous network.

that can result in optimal throughput, while only a few RAW configurations leading to high energy efficiency. Moreover, different from our original version of MoROA that only focuses on throughput, by using the built surrogate model for RAW and multi-objective approach, the extended MoROA is able to either improve throughput up to 5.3%, or improve energy energy efficiency by 32.4% at only a minor 3.26% throughput loss, compared to the state of the art E-TAROA algorithm. Most importantly, MoROA supports traffic-aware RAW optimization for heterogeneous scenarios with variable MCS and packets size. Its objective weight allows to attain either a 65% throughput increase, or a 96% energy saving, or a weighted solution in between compared to the traditional EDCA/DCF channel access method..

In future work, we aim to further extend the surrogate modeling approach to support stations with different MCS and average packet sizes not only across multiple groups, but also within a single RAW group. This would further increase the flexibility of the algorithm in finding an optimal RAW configuration. Moreover, more advanced clustering metrics will be evaluated and compared for use with MoROA. Finally, we will investigate AID reassignment methods to allow MoROA to satisfy the sequential AID requirement of the 802.11ah standard, and take the co-existence of multiple APs into account.

Acknowledgement

Part of this research was funded by the Flemish FWO SBO S004017N IDEAL-IoT (Intelligent DEnse And Long range IoT networks) project. Serena Santi is funded by the Fund For Scientific Research (FWO) Flanders under grant number 1S82118N.

[1] E. Khorov, A. Lyakhov, A. Krotov, A. Guschin, A survey on IEEE 802.11ah: An enabling networking technology for smart cities, Computer Communications 58 (2015) 53–69. doi:10.1016/j.comcom.2014.08.008.

[2] L. Tian, E. Khorov, S. Latr, J. Famaey, Real-time station grouping under dynamic traffic for ieee 802.11ah, Sensors 17 (7).
[3] L. Tian, J. Famaey, S. Latré, Evaluation of the IEEE 802.11ah restricted access window mechanism for dense IoT networks, in: IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016. doi:10.1109/WoWMoM.2016.7523502.
[4] E. Khorov, A. Krotov, A. Lyakhov, Modelling machine type communication in ieee 802.11ah networks, IEEE International Conference on Communication Workshop (ICCW) (14) (2015) 1149–1154. doi:10.1109/ICCW.2015.7247332.
[5] Y. Wang, Y. Li, K. K. Chai, Y. Chen, J. Schormans, Energy-aware adaptive restricted access window for IEEE 802.11ah based smart grid networks, in: IEEE International Conference on Smart Grid Communications (SmartGridComm), 2015, pp. 581–586. doi:10.1109/SmartGridComm.2015.7436363.
[6] L. Tian, M. Mehari, S. Santi, S. Latré, E. De Poorter, J. Famaey, IEEE 802.11ah restricted access window surrogate model for real-time station grouping, in: IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2018.
[7] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, K. Crombecq, A surrogate modeling and adaptive sampling toolbox for computer based design, J. Mach. Learn. Res. 11 (2010) 2051–2055.
URL http://dl.acm.org/citation.cfm?id=1756006.1859919
[8] L. Tian, S. Deronne, S. Latré, J. Famaey, Implementation and validation of an IEEE 802.11ah module for ns-3, in: Proceedings of the Workshop on Ns-3 (WNS3), 2016, pp. 49–56. doi:10.1145/2915371.2915372.
[9] O. Raeesi, J. Pirskanen, A. Hazmi, T. Levanen, M. Valkama, Performance evaluation of IEEE 802.11ah and its restricted access window mechanism, in: IEEE International Conference on Communications Workshops (ICC), 2014, pp. 460–466. doi:10.1109/ICCW.2014.6881241.
[10] A. Lyakhov, E. Khorov, R. Yusupov, Two-slot based model of the IEEE 802.11ah restricted access window with enabled transmissions crossing slot boundaries, in: IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2018.
[11] L. Zheng, M. Ni, L. Cai, J. Pan, C. Ghosh, K. Doppler, Performance analysis of group-synchronized dcf for dense IEEE 802.11 networks, IEEE Transactions on Wireless Communications 13 (11) (2014) 6180–6192. doi:10.1109/TWC.2014.2337315.
[12] C. W. Park, D. Hwang, T.-J. Lee, Enhancement of IEEE 802.11ah MAC for M2M communications, IEEE Communications Letters 18 (7) (2014) 1151–1154. doi:10.1109/LCOMM.2014.2323311.
[13] A. Bel, T. Adame, B. Bellalta, J. Barcelo, J. Gonzalez, M. Oliver, CAS-based channel access protocol for IEEE 802.11ah WLANs, in: Proceedings of 20th European Wireless Conference, 2014.
[14] T. Chang, C. Lin, K. C. Lin, W. Chen, Traffic-aware sensor grouping for ieee 802.11ah networks: Regression based analysis and design, IEEE Transactions on Mobile Computing (2018) 1–1doi:10.1109/TMC.2018.2840692.
[15] S. G. Yoon, J. O. Seo, S. Bahk, Regrouping algorithm to alleviate the hidden node problem in 802.11ah networks, Computer Networks 105 (2016) 22–32. doi:10.1016/j.comnet.2016.05.011.
[16] W. Damayanti, S. Kim, J.-H. Yun, Collision chain mitigation and hidden device-aware grouping in large-scale IEEE 802.11ah networks, Computer Networks 108 (2016) 296–306. doi:10.1016/j.comnet.2016.09.006.
[17] T.-C. Chang, C.-H. Lin, K. C.-J. Lin, W.-T. Chen, Load-balanced sensor grouping for IEEE 802.11ah networks, in: IEEE Global Communications Conference (GLOBECOM), 2015. doi:10.1109/GLOCOM.2015.7417476.
[18] K. Ogawa, M. Morikura, K. Yamamoto, T. Sugihara, IEEE 802.11ah based M2M networks employing virtual grouping and power saving methods, IEICE Transactions on Communications E96-B (12) (2013) 2976–2985.
[19] M. Qutab-ud din, A. Hazmi, B. Badihi, A. Larmo, J. Torsner, M. Valkama, Performance analysis of iot-enabling IEEE 802.11ah technology and its raw mechanism with non-cross slot boundary holding schemes, in: IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015. doi:10.1109/WoWMoM.2015.7158204.
[20] L. Tian, S. Santi, S. Latré, J. Famaey, Accurate sensor traffic estimation for station grouping in highly dense IEEE 802.11ah networks, in: 15th ACM Conference on Embedded Networked Sensor Systems Workshops (SenSys), 2017.
[21] F. Viana, Things you wanted to know about the latin hypercube design and were afraid to ask, in: the 10th World Congress on Structural and Multidisciplinary Optimization, 2013.
[22] I. Couckuyt, T. Dhaene, P. Demeester, oodace toolbox: A flexible object-oriented kriging implementation, Journal of Machine Learning Research 15 (2014) 3183–3186.
URL http://jmlr.org/papers/v15/couckuyt14a.html
[23] J. van der Herten, I. Couckuyt, D. Deschrijver, T. Dhaene, A fuzzy hybrid sequential design strategy for global surrogate modeling of high-dimensional computer experiments, SIAM Journal on Scientific Computing 37 (2)

(2015) A1020–A1039.
[24] T. Van Steenkiste, J. van der Herten, I. Couckuyt, T. Dhaene, Sensitivity analysis of expensive black-box systems using metamodeling, in: Winter Simulation Conference (WSC), 2016, IEEE, 2016, pp. 578–589.
[25] A. Ba, Y.-H. Liu, J. van den Heuvel, P. Mateman, B. Busze, J. Gloudemans, P. Vis, J. Dijkhuis, C. Bachmann, G. Dolmans, K. Philips, H. de Groot, 26.3A 1.3nJ/b IEEE 802.11ah fully digital polar transmitter for IoE applications, in: IEEE International Solid-State Circuits Conference (ISSCC), 2016, pp. 440–441. doi:10.1109/ISSCC.2016.7418096.
[26] A. Ba, K. Salimi, P. Mateman, P. Boer, J. van den Heuvel, J. Gloudemans, J. Dijkhuis, M. Ding, Y.-H. Liu, C. Bachmann, G. Dolmans, K. Philips, A 4mW-RX 7mW-TX IEEE 802.11ah fully-integrated RF transceiver, in: IEEE Radio Frequency Integrated Circuits Symposium (RFIC), 2017, pp. 232–235. doi:10.1109/RFIC.2017.7969060.
[27] R. T. Marler, J. S. Arora, Survey of multi-objective optimization methods for engineering, Structural and multidisciplinary optimization 26 (6) (2004) 369–395.