IEEE 802.11ah Restricted Access Window Surrogate Model for Real-Time Station Grouping

Le Tian*, Michael Mehari†, Serena Santi*, Steven Latré*†,
Eli De Poorter†, Jeroen Famaey*
*University of Antwerp - imec, IDLab, Department of Mathematics and Computer Science, Belgium
†Ghent University - imec, IDLab, Department of Information Technology, 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, a surrogate model that predicts RAW performance given specific network conditions and RAW configuration parameters. It is fast to train and can be solved in real-time. Second, the Model-Based RAW Optimization Algorithm (MoROA), which uses the surrogate model to determine the optimal RAW configuration in real-time, for heterogeneous stations and dynamic traffic. 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 shows that the trained surrogate model can accurately predict RAW performance with a relative error less than 7% and 10% for 95% and 98% of the RAW configurations respectively. MoROA achieves a throughput up to twice as high as traditional 802.11 channel access functions in dense heterogeneous networks.**

I. 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 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. For a more in-depth description of RAW, the reader is referred to existing literature [1], [2].

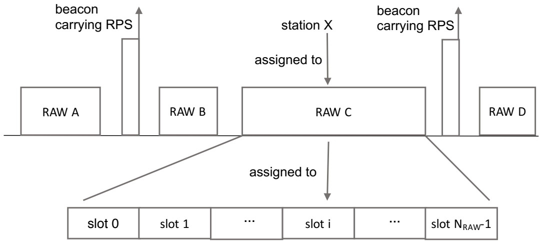


Fig. 1: Schematic representation of the RAW mechanism.

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 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

978-1-5386-4725-7/18/$31.00 ©2018 IEEE

predict RAW performance [4], [5]. However, such models are too computationally hard to be used in real-time, and rely on simplifications and unrealistic assumptions (e.g., no capture effect, no hidden nodes, homogeneous stations, saturated or static traffic). As a first contribution, we present an alternative solution to RAW performance modeling, based on surrogate modeling and trained using realistic simulation results. 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 [6]. This is important, as a RAW configuration depends on many input variables that can take a wide range of values. Moreover, once trained, evaluating the model is equivalent to a constant-time table lookup, which can be easily executed in real-time. As there is no IEEE 802.11ah hardware available, training is done using the realistic IEEE 802.11ah ns-3 model [7]. It implements the capture effect, hidden stations, as well as heterogeneous stations and traffic loads.

As a second contribution, we propose the Model-Based RAW Optimization Algorithm (MoROA). It clusters stations into groups based on their traffic characteristics and determines the optimal RAW configuration by solving a non-linear constrained optimization problem. In this problem, the trained surrogate model is used to maximize throughput and fairness in terms of packets delivery ratio. In contrast to existing algorithms, MoROA supports dynamic and heterogeneous traffic, as well as heterogeneous stations with different modulation and coding schemes (MCSs) and packet sizes [2], [8].

The remainder of this paper is structured as follows. Section II surveys related work in terms of RAW performance modeling and optimization algorithms and compares them to our contributions. Section III details the methodology used to define and train the surrogate model. MoROA is described in Section IV. Section V evaluates the accuracy of our presented model, comparing it to simulation results. 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. Finally, Section VI offers conclusions and a short overview of future work.

II. 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], and maximum likelihood estimation [11]. Early works assume the network is operating under saturated state, where each station always has packets to send [10], [11]. 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 [10]. Park *et al.* determine the RAW group duration for a certain number of stations to get maximized successful transmission probability [11]. In contrast, more recent works assume each station sends one packet per RAW slot interval [4], [5], [12]. 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].

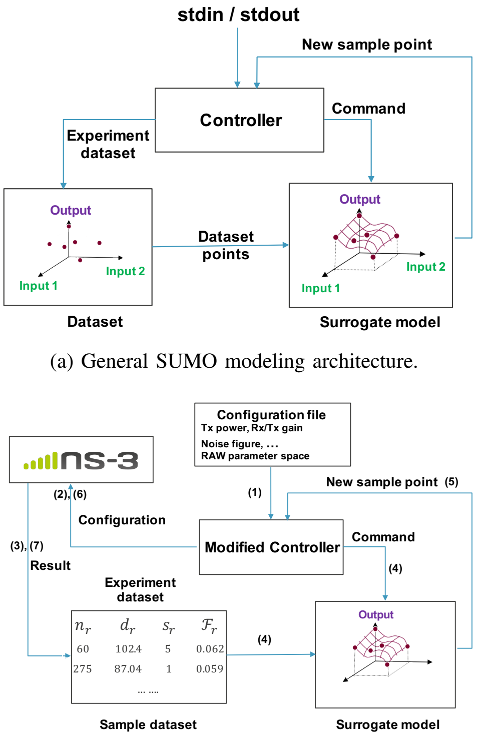
All existing analytical models share two main shortcomings. First, they are computationally hard. This makes it infeasible to execute them in real-time on actual AP hardware, where at most a few milliseconds are available at the start of the beacon interval to calculate a new RAW configuration. More importantly, they assume ideal channel conditions, without communication errors, delays or capture effects. 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 [13], [14]. 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 [15]. Other existing algorithms focus on simple partitioning metrics, such as fully random [16] or based on the back-off timer value [17], 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.

Recently, we proposed the Traffic-Aware RAW Optimization Algorithm (TAROA) [2], [8]. 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 still has two shortcomings that can be addressed. First, it derives the optimal number of stations to assign to a group based on saturated state simulation results. Second, it only supports homogeneous stations (i.e., all stations use the same MCS and average packet size). In this paper we present an improved algorithm, called MoROA, which supports a wide range of traffic conditions and heterogeneous stations, by using generic and flexible surrogate models. This results in significant performance improvements, especially under non-saturated conditions, which are prevalent in IoT and MTC scenarios.

III. 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 of the IEEE 802.11ah RAW under a wide range of network and traffic conditions.



(a) General SUMO modeling architecture.

n_r	d_r	s_r	$\mathcal{F}_r$
60	102.4	5	0.062
275	87.04	1	0.059
... .			

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

Fig. 2: Training methodology used to model RAW performance.

A. 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). Based on this model, optimal RAW parameters can subsequently be derived based on current network conditions. However, existing analytical models are computationally expensive and unrealistic due to their assumptions. Surrogate modeling provides the answer [6]. 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.

The Matlab Surrogate Modeling (SUMO) Toolbox is a flexible framework for accurate global surrogate modeling [6]. 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 [7]. 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 I) and the input space parameters (cf. Table II).
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) 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 built surrogate model estimates the next sample data point to evaluate with the highest expected accuracy improvement.
6) The controller starts the next ns-3 experiment for the newly selected sample data point.
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 [18] to generate 100 initial sample data points, Kriging interpolation is used to train the model [19], FLOLA-Voronoi sampling for generating the next sample points [20], and the 10-fold cross-validation with a root-relative square error (RRSE) measure to evaluate the model accuracy [21]. The training stops once the cross-validation score lower than or equal to 0.10 (2 digits of precision) occurs 10 times in succession, or the number of training data points exceeds 2500.

B. 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 I. 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. [22], 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 optimization algorithm proposed in Section IV 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 to be trained for each MCS that stations are expected to use. We illustrate this by developing a separate a 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 IV 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.

TABLE I: 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
High-Throughput (HT) parameters	**value**
Wi-Fi mode	MCS1, 1 Mhz
Average payload size (bytes)	64
Topology radius (m)	200
Low-Throughput (LT) parameters	**value**
Wi-Fi mode	MCS9, 1 Mhz
Average payload size (bytes)	256
Topology radius (m)	80

C. Input parameters for RAW modeling

The surrogate model aims to accurately predict throughput t_r, 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 a function $\mathcal{F}$, as follows:

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

In addition to throughput, the simulator calculates a variety of other performance metrics, such as packet loss, latency and energy consumption. 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.

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 I, a minimum value of 60 and maximum value of 400 stations per group were deemed to cover all possible traffic conditions.

TABLE II: 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

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

IV. 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) [8]. As such, it is also traffic-aware and able to adapt to changing traffic conditions. In contrast to previous work, MoROA differentiates in using a model to find the optimal RAW configuration parameters. This allows it to better estimate the actual performance of a specific RAW configuration. Moreover, MoROA supports heterogeneous stations with variable MCS and packet size.

A. Overview

As in Section III, 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 latency).

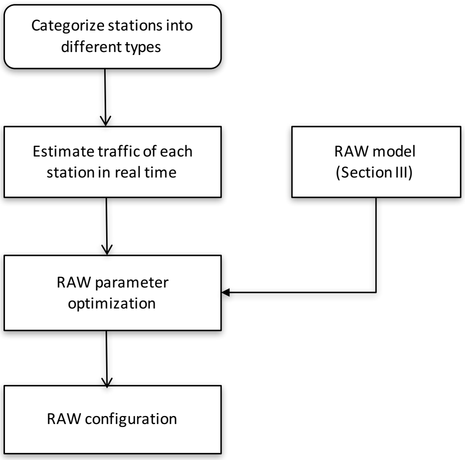

Categorize stations into different types
Estimate traffic of each station in real time
RAW model (Section III)
RAW parameter optimization
RAW configuration

Fig. 3: Overview of MoROA.

The proposed algorithm uses only information readily available at the AP. The concrete steps are illustrated in Figure 3. 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 [8]. Finally, we utilize a RAW performance model function $\mathcal{F}(n, d, s)$ that takes 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). 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 metric. 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 III. However, any function $\mathcal{F}(n, d, s)$ that satisfies the above requirements and that can be calculated in real-time could be used in combination with MoROA.

B. 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 [8]. 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 maximize the chosen performance metric can be formulated as an optimization problem. This problem has to be solved jointly for all groups, as the chosen duration d_i of a cluster i influences the maximum duration of all other groups (i.e., the sum of all group durations should not be higher than the beacon interval duration d_b).

As stated in Section III, a variety of objective functions can be defined, as the model can be trivially trained for a variety of metrics such as throughput, latency, energy consumption, and packet loss. As an illustration, we define an objective function that maximizes throughput as well as fairness across groups in terms of packets delivery ratio. The packet delivery ratio is defined as the ratio between the total number of packets that are successfully received and the total number of packets generated. However, other objectives such as airtime fairness, latency minimization, or energy efficiency can easily be defined as well. The problem can be formulated as follows:

$$\max\left(\alpha \times \mathcal{Q}_p + (1-\alpha) \times \mathcal{Q}_f\right) \tag{2}$$

With:

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

And

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

Subject to:

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

$$\sum_{i=1}^{k} d_i \leq d_b \tag{6}$$

$$\forall i \in [1,k] : p_i^s < \frac{n_i \times l_i}{\mathcal{F}_t\left(n_i, d_i, s_i\right)} \tag{7}$$

Where $\mathcal{F}_t\left(\cdot\right)$ represents the RAW model function that calculates throughput. 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. 7, 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_f represents the fairness objective, while Q_p represents throughput. Both objectives are normalized as to align their valid value range and simplify selection of α. 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 [8]. However, this has a negligible effect on performance of the algorithm over longer periods. Currently, MoROA does not consider the required sequentiality 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, reducing 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. The number of groups k cannot be higher than 42, as the maximum size of the RPS elements is 256 bytes and at least 6 bytes are needed per group. Based on the specification of RAW, the parameter s_i is at most 64 and d_i is at most 1990.4 ms.

V. 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.

A. simulation setup

All evaluations are performed using our previously developed IEEE 802.11ah ns-3 module [7], based on ns-3 version 3.23. We consider the same IoT scenario as described in

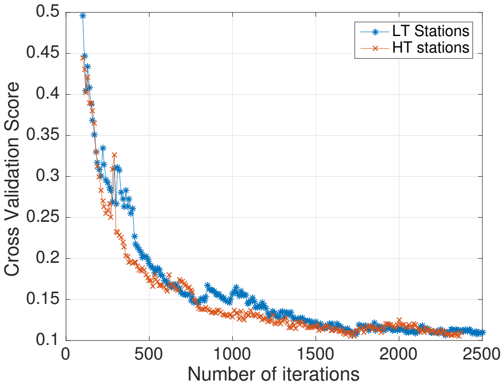


Fig. 4: Cross validation score of the surrogate model as a function of the number of training iterations.

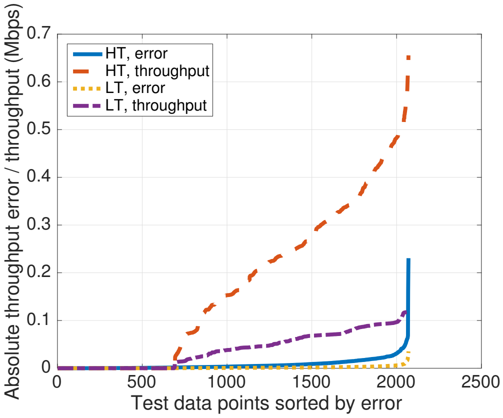


Fig. 5: Performance comparison between the surrogate model and simulation results for 2070 random test data points, sorted in ascending order in terms of estimation error.

Section IV. The same default PHY and MAC layer parameters used as shown in Table I. We consider both homogeneous and heterogeneous scenarios. Homogeneous scenarios are used to validate the surrogate model and compare to E-TAROA [8]. In heterogeneous scenarios, half of the stations use the high-thoughput (HT) settings and half of them use the low-throughput settings (LT) listed in Table I. The data transmission interval of each station is selected uniformly at random from the interval $[1, 10]$ seconds.

RAW performance is evaluated in terms of three metrics: throughput, fairness, and latency. Throughput is calculated as the average number of successfully received payload bytes by the AP per second. Fairness between HT and LT stations in terms of packet delivery ratio is evaluated using Jain's fairness index. Latency is defined as the average time between a packet entering the transmit queue of the station and being received by the AP. 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).

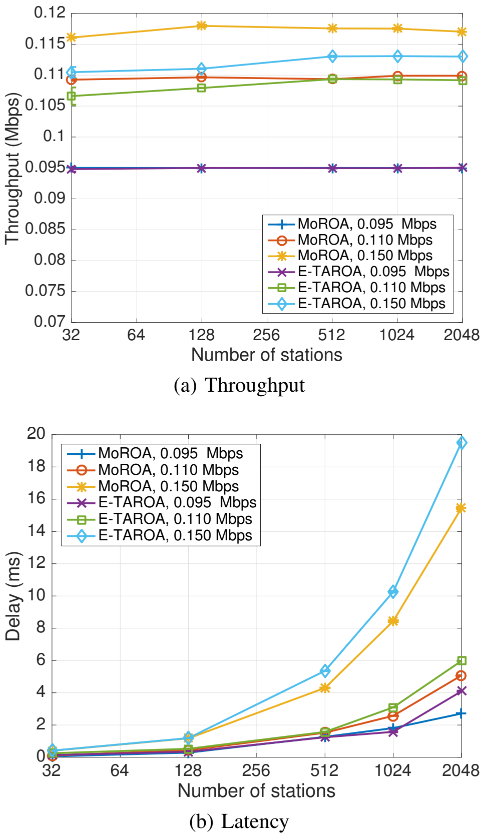
B. SUMO model validation

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. A consistently low score signifies that the training process has converged. Based on this graph, we can conclude that for both the model with high-throughput (HT) and low-throughput (LT) stations, convergence occurs after around 1700 training samples have been used. The training of the HT model stopped after 2370 iterations as it satisfied the cross-validation stop conditions, i.e., 10 consecutive cross-validation scores lower than or equal to 0.10 (2 digits of precision). The training of the LT model stopped after the maximum number of 2500 iterations, having only achieved 15 consecutive cross-validation scores between 0.11 and 0.12. 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.

In order to ensure no over-fitting occurred, the surrogate model also provides accurate results for data points outside the reduced input space (cf. Table II), Figure 5 plots the absolute error (in terms of throughput) of the surrogate model compared to simulation results. In total, 2070 random data points were generated from all 2.3×10^7 possible points. The figure also plots the actual simulated throughput of each point, to characterize the significance of the error. For the HT model, the absolute throughput estimation error stays below 0.02 Mbps for the first 2000 points (i.e., 96.6% of all data points). The absolute error only goes above 0.1 Mbps for 3 data points (i.e., 0.14%). Similarly, for the LT model, the throughput estimation error stays below 0.004 Mbps for the first 2000 data points, and only grows above 0.01 Mbps for the worst 11 data points. In terms of the relative error (i.e., ratio between absolute throughput error and simulation results, not depicted) of the surrogate model compared to simulations, 95% of all data points have an error below 6.6% and 5.4% for the HT and LT model respectively. For both models, 98% of the data points have a relative error below 10%. 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.

C. Homogeneous stations

In this section we evaluate the performance of MoROA for homogeneous stations (in terms of MCS and packet size) for a variety of traffic loads and station counts. Three different total traffic loads are simulated for the LT scenario, i.e., $T = \{0.095, 0.11, 0.15\}$ Mbps. Given the packet payload size and data rate, the maximum throughput that can be achieved is about 0.124 Mbps for LT stations. As such, $T = 0.15$ Mbps represents a near-saturated state, $T = 0.11$ Mbps represents a medium traffic load and $T = 0.095$ Mbps results in low traffic load. As E-TAROA was already shown to outperform



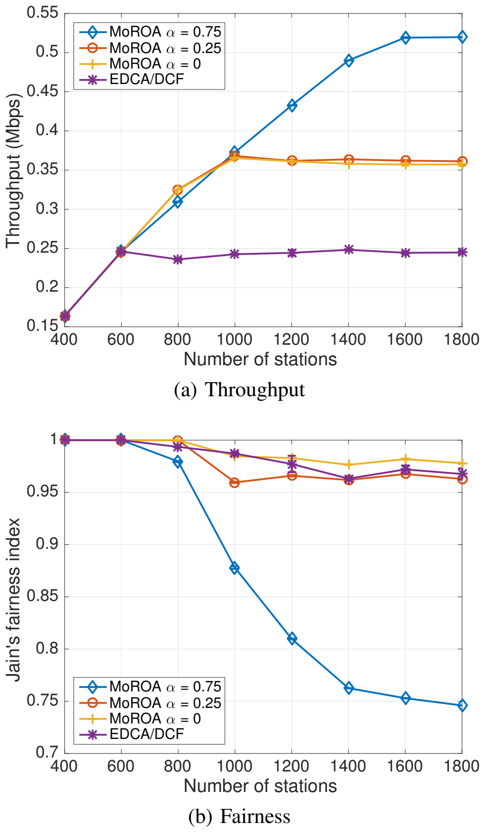
(a) Throughput

(b) Latency

Fig. 6: Performance comparison between MoROA and E-TAROA for the LT scenarios with different traffic loads and station counts.

EDCA/DCF for homogeneous stations, we use E-TAROA as a benchmark in this section [8].

Figure 6 depicts performance (i.e., throughput and latency) of MoROA and E-TAROA. It clearly shows that, compared to E-TAROA, MoROA improves the throughput under near saturated traffic conditions for any number of stations. At low traffic loads, MoROA and E-TAROA have the same throughput for any number of stations. Under medium traffic conditions, MoROA also improves performance for a low number of stations. More importantly, better latency performance is achieved in denser networks. For a low traffic load (i.e., 0.095 Mbps), MoROA and E-TAROA have nearly the same latency when there are less than 1024 stations. However, for 2048 stations, latency of MoROA is about 32.5% lower. For higher traffic loads, MoROA has better latency even in less dense networks (i.e., starting from 512 and 128 stations for traffic load 0.11 and 0.15 Mbps respectively). The above results reveal that, by using surrogate modeling, stations are assigned to RAW groups in a more optimal way. This is due to the fact that E-TAROA derives its optimal RAW parameters from a model under saturated state as an approximation, rather than optimizing them for different traffic conditions as the surrogate model does. This results in better overall performance in terms of throughput and latency.



(a) Throughput

(b) Fairness

Fig. 7: Performance comparison between MoROA and EDCA/DCF, for a heterogeneous network.

D. 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 fairness among the LT and HT station groups, for a variety of network densities. As E-TAROA only supports stations with the 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 7. With a small number of stations, MoROA and EDCA/DCF have the same throughput. With more than 600 stations, the throughput of EDCA/DCF stagnates to about 0.25 Mbps as the network becomes saturated. However, MoROA reduces the number of collisions, resulting in better scaling behavior and a higher saturation limit. As a result, for 1800 stations throughput of MoROA is about 44% higher for $\alpha = 0$ and $\alpha = 0.25$. If more focus is put on the throughput objective, with $\alpha = 0.75$, MoROA more than doubles throughput compared to EDCA/DCF in highly dense networks (i.e., 1800 stations), achieving a throughput increase of 108%. As expected, EDCA/DCF achieves high fairness among HT and LT stations due to its random access mechanism. For $\alpha \leq 0.25$ MoROA achieves a similar degree of fairness. However, for $\alpha = 0.75$ fairness of MoROA is significantly reduced, as it prioritizes HT stations to maximize overall throughput.

VI. 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 a new RAW performance model, based on supervised surrogate modeling. It can be easily trained on a limited set of labeled data samples, which can be obtained through simulation. Moreover, it is very fast to evaluate once trained, allowing it to be used for real-time RAW parameter optimization. The second contribution encompasses a RAW optimization algorithm called MoROA. It uses the surrogate model to determine the optimal RAW configuration under a variety of network and traffic conditions. Moreover, it supports heterogeneous stations with different MCS and average packet sizes.

The simulation results reveal three 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 6.6% for 95% of the randomly tested RAW configurations is achieved. Second, by using the built surrogate model for RAW, MoROA achieves more stable throughput for both low- and high-density deployments as well as up to 32.5% lower latency, compared to the state of the art TAROA algorithm. Most importantly, MoROA supports traffic-aware RAW optimization for heterogeneous scenarios with variable MCS and packets size, achieving up to twice the throughput of EDCA/DCF in dense networks.

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.

References

[1] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on IEEE 802.11ah: An enabling networking technology for smart cities," *Computer Communications*, vol. 58, pp. 53–69, 2015.

[2] L. Tian, E. Khorov, S. Latr, and J. Famaey, "Real-time station grouping under dynamic traffic for ieee 802.11ah," *Sensors*, vol. 17, no. 7, 2017.

[3] L. Tian, J. Famaey, and 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.

[4] E. Khorov, A. Krotov, and A. Lyakhov, "Modelling machine type communication in ieee 802.11ah networks," *IEEE International Conference on Communication Workshop (ICCW)*, no. 14, pp. 1149–1154, 2015.

[5] Y. Wang, Y. Li, K. K. Chai, Y. Chen, and 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.

[6] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A surrogate modeling and adaptive sampling toolbox for computer based design," *J. Mach. Learn. Res.*, vol. 11, pp. 2051–2055, Aug. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1859919

[7] L. Tian, S. Deronne, S. Latré, and 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.

[8] L. Tian, S. Santi, S. Latré, and 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.

[9] O. Raeesi, J. Pirskanen, A. Hazmi, T. Levanen, and 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.

[10] L. Zheng, M. Ni, L. Cai, J. Pan, C. Ghosh, and K. Doppler, "Performance analysis of group-synchronized dcf for dense IEEE 802.11 networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 11, pp. 6180–6192, 2014.

[11] C. W. Park, D. Hwang, and T.-J. Lee, "Enhancement of IEEE 802.11ah MAC for M2M communications," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1151–1154, 2014.

[12] A. Bel, T. Adame, B. Bellalta, J. Barcelo, J. Gonzalez, and M. Oliver, "CAS-based channel access protocol for IEEE 802.11ah WLANs," in *Proceedings of 20th European Wireless Conference*, 2014.

[13] S. G. Yoon, J. O. Seo, and S. Bahk, "Regrouping algorithm to alleviate the hidden node problem in 802.11ah networks," *Computer Networks*, vol. 105, pp. 22–32, 2016.

[14] W. Damayanti, S. Kim, and J.-H. Yun, "Collision chain mitigation and hidden device-aware grouping in large-scale IEEE 802.11ah networks," *Computer Networks*, vol. 108, pp. 296–306, 2016.

[15] T.-C. Chang, C.-H. Lin, K. C.-J. Lin, and W.-T. Chen, "Load-balanced sensor grouping for IEEE 802.11ah networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2015.

[16] K. Ogawa, M. Morikura, K. Yamamoto, and T. Sugihara, "IEEE 802.11ah based M2M networks employing virtual grouping and power saving methods," *IEICE Transactions on Communications*, vol. E96-B, no. 12, pp. 2976–2985, 2013.

[17] M. Qutab-ud din, A. Hazmi, B. Badihi, A. Larmo, J. Torsner, and 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.

[18] 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.

[19] I. Couckuyt, T. Dhaene, and P. Demeester, "oodace toolbox: A flexible object-oriented kriging implementation," *Journal of Machine Learning Research*, vol. 15, pp. 3183–3186, 2014. [Online]. Available: http://jmlr.org/papers/v15/couckuyt14a.html

[20] J. van der Herten, I. Couckuyt, D. Deschrijver, and T. Dhaene, "A fuzzy hybrid sequential design strategy for global surrogate modeling of high-dimensional computer experiments," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A1020–A1039, 2015.

[21] T. Van Steenkiste, J. van der Herten, I. Couckuyt, and T. Dhaene, "Sensitivity analysis of expensive black-box systems using metamodeling," in *Winter Simulation Conference (WSC), 2016*. IEEE, 2016, pp. 578–589.

[22] 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, and 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.