

Accurate Sensor Traffic Estimation for Station Grouping in Highly Dense IEEE 802.11ah Networks

Le Tian

IDLab, University of Antwerp – imec
Antwerp, Belgium
Le.Tian@uantwerpen.be

Steven Latré

IDLab, University of Antwerp – imec
Antwerp, Belgium
Steven.Latre@uantwerpen.be

Serena Santi

IDLab, University of Antwerp – imec
Antwerp, Belgium
Serena.Santi@uantwerpen.be

Jeroen Famaey

IDLab, University of Antwerp – imec
Antwerp, Belgium
Jeroen.Famaey@uantwerpen.be

ABSTRACT

The restricted access window (RAW) feature of IEEE 802.11ah aims to significantly reduce channel contention in ultra-dense and large-scale sensor networks. It divides stations into groups and slots, allowing channel access only to one RAW slot at a time. Several algorithms have been proposed to optimize the RAW parameters (e.g., number of groups and slots, group duration, and station assignment), as the optimal parameter values significantly affect performance and depend on network and traffic conditions. These algorithms often rely on accurate estimation of future sensor station traffic. In this paper, we present a more accurate traffic estimation technique for IEEE 802.11ah sensor stations, by exploiting the “more data” header field and cross slot boundary features. The resulting estimation method is integrated into an enhanced version of the Traffic-Adaptive RAW Optimization Algorithm, referred to as E-TAROA. Simulation results show that our proposed estimation method is significantly more accurate in very dense networks with thousands of sensor stations. This in turn results in a significantly more optimal RAW configuration. Specifically, E-TAROA converges significantly faster and achieves up to 23% higher throughput and 77% lower latency than the original TAROA algorithm under high traffic loads.

CCS CONCEPTS

• **Networks** → **Network dynamics**; **Network management**; **Wireless access networks**; • **Computing methodologies** → **Modeling and simulation**;

KEYWORDS

IEEE 802.11ah, highly dense sensor networks, traffic estimation, restricted access window (RAW), optimal station grouping

ACM Reference format:

Le Tian, Serena Santi, Steven Latré, and Jeroen Famaey. 2017. Accurate Sensor Traffic Estimation for Station Grouping in Highly Dense IEEE 802.11ah Networks. In *Proceedings of ACM Conference on Embedded Networked Sensor Systems, Delft, The Netherlands, November 2017 (SENSYS’17)*, 8 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

The Internet of Things (IoT) ultimately envisions connecting tens of billions of low-power and resource-constrained devices to the Internet. This will result in ultra-dense deployments of sensors and actuators, with thousands of them coexisting in a small geographical area. Enabling this vision will require novel wireless communications solutions that scale properly to such extreme proportions. It is widely known that traditional media access control (MAC) scheduling methods, such as carrier sense multiple access (CSMA), time division multiple access (TDMA), and ALOHA do not scale properly under ultra-dense conditions. CSMA has been shown to result in significant overall throughput degradation due to contention with only a few hundred stations connected to the access point (AP) [13]. Although TDMA scales better as the network becomes denser, it is less flexible and results in wasted airtime in heterogeneous and low-throughput environments [13]. Finally, ALOHA has an even higher collision probability than CSMA in dense environments. Recent measurements of LoRaWAN, which employs an ALOHA-like MAC, show that as few as 100 stations on a small geographical area already results in significantly reduced performance [3].

The recently released long-range and low-power Wi-Fi standard IEEE 802.11ah proposes a novel channel access method, referred to as the 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 AP. The 802.11ah standard, however, does not specify how to configure the actual RAW grouping parameters. These parameters include the number of groups, the duration of each group, the number of (equal-sized) slots in each group, and which stations to assign to each group.

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

SENSYS’17, November 2017, Delft, The Netherlands

© 2017 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Previously, we conducted an in-depth analysis of station grouping, and concluded that the optimal RAW parameters depend on a wide range of network variables, such as number of stations, network load and traffic patterns [13]. Incorrect configuration severely impacts throughput, latency and energy efficiency. Several algorithms have been proposed to determine suitable RAW parameters. For sensor network traffic with either 1 packet per station or under saturation, some analytical models were proposed. These models are based on different techniques, such as probability theory [10, 15], Markov chains [7, 16], multi-objective game theory [2], and maximum likelihood estimation [9]. However, these models are computationally hard, which makes it infeasible to execute them in real-time on actual AP hardware. As an alternative that is computationally feasible and deployable, several partitioning algorithms were proposed. They partition the stations into different RAW slots based on different metrics, such as arbitration inter-frame space number (AIFSN) value [8], and station traffic load [4]. However, this information is not known to the AP in reality, also making them infeasible to implement. Recently, we proposed a real-time station grouping algorithm, named TAROA, by estimating the traffic conditions of each station with information only available at the AP [14]. In contrast to other state-of-the-art algorithms it is capable of adjusting its RAW configuration in real-time, in face of station and traffic dynamics.

In this paper, we propose a more accurate traffic estimation technique for TAROA by exploiting the “more data” header field and cross slot boundary features of IEEE 802.11ah. By using the “more data” field of the data packet, a station can indicate to the AP that its transmission queue still has packets, the cross slot boundary feature allows a station to continue an ongoing transmission after the end of the current RAW slot. The proposed traffic estimation approach can quickly adjust the estimation and avoid false negative packet transmissions for estimation. The estimation method is integrated into TAROA and referred to as Enhanced TAROA (E-TAROA). Thorough evaluations in the extended version of 802.11ah ns-3 simulator [11] are performed to compare E-TAROA to the original algorithm in terms of traffic estimation accuracy, throughput, latency and energy efficiency.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of the RAW feature. The more accurate sensor traffic estimation approach and E-TAROA are described in Section 3. In Section 4, we provide the simulation results of E-TAROA and compare it to the original TAROA algorithm. Finally, Section 5 concludes the paper.

2 RESTRICTED ACCESS WINDOW

The RAW mechanism aims to mitigate collisions and improve performance in dense sensor networks in which a large number of stations are contending for channel access simultaneously. It splits stations into groups and only allows stations assigned to a certain group to access the channel at specific times. 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 CSMA with collision avoidance (CSMA/CA) method. At fixed intervals a beacon frame is transmitted, carrying a

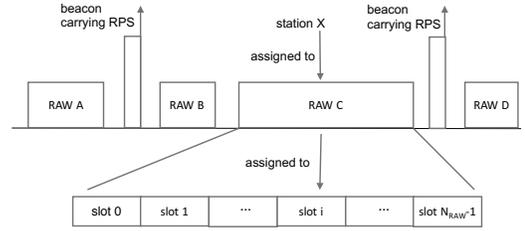


Figure 1: Schematic representation of the RAW mechanism, with the beacon RPS element carrying information about the number of RAW groups, their duration, number of equal-sized slots and assigned stations

RAW parameter set (RPS) information element, which specifies the stations belonging to each group, 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 as follows:

$$D = 500 \mu s + C \times 120 \mu s \quad (1)$$

Where C represents the *slot duration count* sub-field, which is either $y = 11$ or $y = 8$ bits long if the slot format sub-field is set to respectively 1 or 0. The *number of slots* field is $14 - y$ bits long. As such, if $y = 11$, each RAW group consists of at most 8 slots and the slot duration is up to 246.14 ms. If $y = 8$, each RAW group consists of at most 64 slots and the slot duration is limited to 31.1 ms at most. Stations are mapped to slots as follows:

$$i_{slot} = (x + N_{offset}) \bmod N_{RAW} \quad (2)$$

Where i_{slot} is the index of i^{th} RAW slot to which the station is mapped. N_{RAW} is the number of slots in one RAW group. N_{offset} is the offset value in the mapping function to improve fairness and equals the two least significant octets of the *FCS field* of the S1G beacon frame, and x is the index of the station. Figure 1 shows an example of the RAW slot assignment procedure.

The RPS also contains the *cross slot boundary* (CSB) sub-field. As Figure 2 depicts, stations are allowed to continue their ongoing transmissions even after the end of the current RAW slot when CSB is set to true. Otherwise, stations should not start a transmission if the remaining time in the current RAW slot is not enough to complete it. The remaining time, termed as “holding time”, should be at least equal to the TXOP of the station. Hazmi et al. [5] proposed several holding schemes that specify how the station should count its back-off within the holding period.

3 ENHANCED TAROA

This section describes the Enhanced Traffic-Adaptive RAW Optimization Algorithm (E-TAROA), which enhances the traffic estimation method of the TAROA algorithm [14]. First, an overview is given of TAROA and its enhanced version. Subsequently, the derivation of packet transmission interval information by exploiting the “more data” header field and cross slot boundary features is

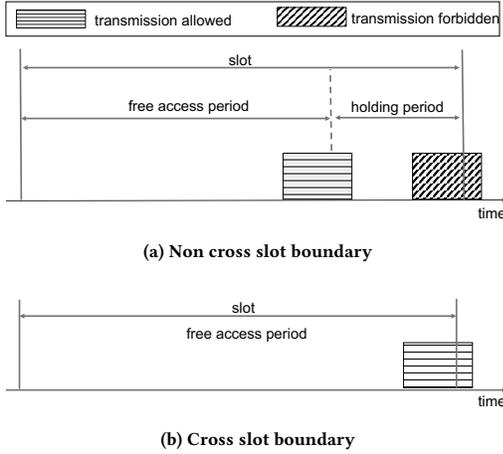


Figure 2: Illustration of the non cross slot boundary and cross slot boundary features of RAW mechanism in IEEE 802.11ah

Table 1: Variables introduced in the algorithm description

Variable name	Description
t_c	Current time*
$\pi_{b,r}^s$	Number of packets received by the AP in RAW slot r of beacon interval b
π_{failed}^s	Number of consecutive failed transmissions
t_{next}^s	Estimated next transmission time*
t_{int}^s	Estimated transmission interval*
\hat{t}_{int}^s	Real transmission interval*
$t_{succ}^s[0]$	Last successful transmission time*
$t_{succ}^s[1]$	Previous to last successful transmission time*
$\pi_{trans}^s[0]$	Last transmission result, success or failure
$\pi_{trans}^s[1]$	Previous to last transmission result
$m_{succ}^s[0]$	Transmission queue has packets, indicated by last successful transmission packet
$m_{succ}^s[1]$	Transmission queue has packets, indicated by Previous to last successful transmission
c_{succ}^s	Packet transmission span two beacon interval
Δ_m^s	Number packet transmission resulting in more packet in transmission queue

* Expressed as a multiple of number of beacon intervals.

described. Finally, based on this derived information, the proposed novel traffic estimation algorithm is presented. Table 1 provides an overview of the variables used throughout this section.

3.1 General overview

TAROA [14] aims to optimize the RAW parameters in real time, in order to get high performance in ultra-dense and large-scale sensor networks. TAROA targets IoT sensor-based monitoring scenarios, where a large set of sensor stations \mathcal{S} send measurements to a backend server (through the AP) at specific time intervals. A sensor station s usually has a predictable packet transmission interval \hat{t}_{int}^s ,

which may change over time (e.g., when an environmental event triggers a change in the sensor measurement interval).

In TAROA, optimizing the RAW parameters strongly relies on accurate estimation of this interval \hat{t}_{int}^s for each station s . However, TAROA does not fully utilize all the information available in the IEEE 802.11ah packet header, which could provide more insights into the station's traffic pattern. Therefore, we propose a more accurate traffic estimation method by exploiting the "More Data" header field and cross slot boundary feature. We refer to the integration of TAROA and this novel traffic estimation method as Enhanced TAROA (E-TAROA).

TAROA, as well as E-TAROA, is executed at each target beacon transmission time (TBTT) and consists of two main steps. First, the AP applies the proposed traffic estimation method to determine the estimated next transmission time of each stations. Second, the algorithm determines the RAW parameters (number of groups, slots per group, and group duration) and assigns stations to each RAW group based on their estimated traffic.

The first step of E-TAROA (traffic estimation) is significantly different from the version of TAROA, and is described in the remainder of this section. The second step (RAW parameter optimization) of E-TAROA is nearly identical to the one of the original algorithm. The only difference is that E-TAROA allows stations with queued packets, but an estimated next transmission time greater than the time of the next beacon, to transmit, while TAROA did not consider their buffer and would not schedule them for transmission. As such, we refer the reader to the description of the original algorithm for a more detailed overview of this step [14].

3.2 Deriving transmission information

The traffic estimation of each station in TAROA is based on the successful and failed transmissions during the previous beacon interval, which can be directly obtained from the number of packets received by the AP. However, by exploiting the "more data" header field and the cross slot boundary feature of IEEE 802.11ah, additional information about the station's queue and traffic generation interval can be derived and more accurate traffic estimation can be achieved.

IEEE 802.11ah inherits the "more data" header field from the legacy IEEE 802.11 standard and further extends its application scope. An 802.11ah station can set the "more data" header field of the frame control field to 1 in individually addressed packets to indicate that it has packets queued for transmission to the AP. In the legacy 802.11 standard, the "more data" field can be only used in the contention free (CF) period of the Point Coordination Function (PCF) or PS (power save) mode. Figure 3a depicts an example of a station notifying the AP that it has more packets in its queue using the "more data" header field. It shows, when the estimated traffic interval t_{int}^s (i.e., 7 in the example) of station s is larger than the real transmission interval \hat{t}_{int}^s (i.e., 5 in the example), there is one packet buffered at station s after Δ_m^s times transmission. In this case, the real transmission interval \hat{t}_{int}^s satisfies the following conditions:

$$\begin{cases} (t_{int}^s - \hat{t}_{int}^s) \times \Delta_m^s + offset \geq \hat{t}_{int}^s \\ (t_{int}^s - \hat{t}_{int}^s) \times \Delta_m^s + offset < \hat{t}_{int}^s \times 2 \end{cases} \quad (3)$$

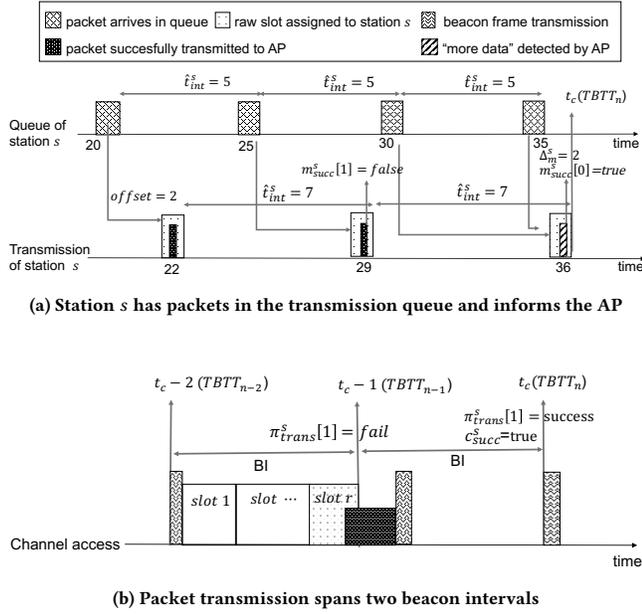


Figure 3: Parameters used for estimating the transmission interval of station s at time t_c

Where *offset* means, right *offset* beacon intervals before t_{int}^s is calculated for the first time, a packet arrives in the transmission queue. *offset* ranges from 0 to $\hat{t}_{int}^s - 1$. Therefore, by using the “more data” header field, the estimated lower and upper bound of the real transmission interval \hat{t}_{int}^s can be represented as:

$$\begin{cases} t_{intMin}^{s,m} = \frac{\Delta_m^s}{\Delta_m^s + 2} \times t_{int}^s + 1 \\ t_{intMax}^{s,m} = t_{int}^s - 1 \end{cases} \quad (4)$$

The cross slot boundary feature allows ongoing transmissions to continue after the end of the current RAW slot. This feature can cause a packet transmission to span two beacon intervals. As a consequence, traffic estimation, which is performed at the TBTT, will consider such a transmission as failed, while in reality it may succeed. An example of this is depicted in Figure 3b. TAROA considers the transmission a failure at time $t_c - 1$ but a success at time t_c . E-TAROA improves estimation accuracy by accounting for the incorrect information used at time $t_c - 1$ and correcting it at time t_c , as shown below.

3.3 Sensor traffic estimation

The goal of sensor traffic estimation is to determine the estimated transmission interval t_{int}^s and next transmission time t_{next}^s for each station s . As shown in Algorithm 1, they are estimated based on successful and failed transmissions during the previous beacon interval. A station’s transmission is regarded as successful if the AP received at least one packet from the station. When only one packet from a station was received, and the station was not assigned to a RAW slot, the AP assumes the transmission crossed two beacon intervals. By using the “more data” field, a successful transmission can further indicate whether the station has more packets in its

Algorithm 1: Transmission interval estimation of station s

```

input :  $t_{succ}^s[0], t_{succ}^s[1], \pi_{trans}^s[0], \pi_{trans}^s[1], t_{int}^s, t_c, \pi_{b,r}^s, \pi_{failed}^s$ 
          $m_{succ}^s[0], m_{succ}^s[1], c_{succ}^s, \Delta_m^s$ 
output:  $t_{int}^s, t_{next}^s$ 

1 if  $\pi_{trans}^s[0] == failed \wedge m_{succ}^s[0] == false$  then
2    $\pi_{failed}^s \leftarrow \pi_{failed}^s + 1$ 
3    $t_{int}^s \leftarrow t_c - t_{succ}^s[0] + 2 \times \pi_{failed}^s - 1$ 
4 else if  $\pi_{trans}^s[0] == success \wedge \pi_{trans}^s[1] == failed$  then
5    $\pi_{failed}^s \leftarrow 0$ 
6   if  $c_{succ}^s == true$  then
7      $t_{succ}^s[0] \leftarrow 1$ 
8      $t_{int}^s \leftarrow t_{succ}^s[0] - t_{succ}^s[1]$ 
9 else if  $\pi_b^s == 1$  then
10   $\pi_{failed}^s \leftarrow 0$ 
11  if  $t_{int}^s > 1 \wedge m_{succ}^s[1] == true \wedge m_{succ}^s[0] == false$ 
    then
12     $t_{intMin}^s = \max(t_{int}^s - 2 \times (\pi_{fail}^s - 1), \frac{\Delta_m^s}{\Delta_m^s + 2} \times t_{int}^s + 1)$ 
13     $t_{intMax}^s = t_{int}^s - 1$ 
14     $t_{int}^s = (t_{intMin}^s + t_{intMax}^s) / 2$ 
15  else if  $t_{int}^s > 1$  then
16     $t_{int}^s \leftarrow t_{succ}^s[0] - t_{succ}^s[1]$ 
17  else if  $c_{succ}^s == false \wedge m_{succ}^s[0] == false$  then
18     $t_{int}^s \leftarrow 1$ 
19 else if  $\pi_b^s > 1$  then
20   $\pi_{failed}^s \leftarrow 0$ 
21  if  $t_{int}^s > 1$  then
22     $t_{int}^s \leftarrow t_{int}^s - 1$ 
23  else if  $\pi_b^s > 1/t_{int}^s$  then
24     $1/t_{int}^s \leftarrow 1/t_{int}^s + 1$ 
25  else if  $\pi_b^s < 1/t_{int}^s \wedge m_{succ}^s[0] == false$  then
26     $1/t_{int}^s \leftarrow 1/t_{int}^s - 1$ 
27  $t_{next}^s = t_{int}^s + t_{succ}^s[0]$ 

```

queue or not. If a station was assigned a RAW slot, but no packets were received by the AP, it is considered a failed transmission. The algorithm consists of four main blocks: (i) the previous transmission failed (lines 1–3), (ii) the previous transmission was successful, but the one before failed (lines 4–8), (iii) the last two transmissions were successful, and only one packet was received in the previous beacon interval (lines 9–18), and (iv) the last two transmissions were successful, and more than one packet was received in the beacon interval (lines 19–26).

In the first case (lines 1–3), if the previous successful transmission did not indicate station s had packets queued to send, and the previous transmission failed, the transmission failure counter π_{failed}^s is increased by 1, and t_{int}^s is increased by the number of subsequent failed transmissions multiplied by two. The transmission failure can be caused by a collision or the lack of packets in the station’s transmission queue (i.e. no packets arrived in the transmission

queue since the previous successful transmission). As RAW aims to minimize collisions, by properly grouping the stations into RAW slots, we assume the probability of transmission failure caused by collision is low enough to be ignored. Thus, the estimated transmission interval t_{int}^s of station s is considered too short. As shown in Line 3, as the number of sequential failed transmission attempts increases, the algorithm assumes its estimation is more wrong and it will increase the interval faster. As such, the transmission interval can be overestimated up to $2 \times (\pi_{fail}^s - 1)$ beacon intervals. The first case also implies that if the failed transmission was caused by collision (i.e., the station had buffered packets to send but the previous transmission failed), t_{int}^s remains unchanged.

In the second case (lines 4–8), the failure counter π_{failed}^s is set to 0 and the transmission interval is estimated as the time difference between the last two successful transmissions, i.e. $t_{succ}^s[0]$ and $t_{succ}^s[1]$. If the previous successful transmission crosses two beacon intervals, $t_{succ}^s[0]$ is decreased by one beacon interval.

If an accurate t_{int}^s is obtained in case 2, the next transmission will succeed and lead to case 3 (lines 9–18) or case 4 (lines 19–26) in which the two last transmissions are successful. In case 3, only 1 packet was received in the previous beacon interval. In both cases 3.1 (lines 11–14) and 3.2 (lines 15–16), t_{int}^s is larger than one. For case 3.1, the transmission queue had no packets at the time of the last successful transmission but had packets at the time of previous to last successful transmission. This means the transmission interval was overestimated, and after Δ_m^s times transmission, this overestimation results in one buffered packet in the transmission queue at the time of the previous to last successful transmission. The estimated lower bound of the real transmission interval is updated as the maximum of the lower bound depicted in Equation 4, and $t_{int}^s - 2 \times (\pi_{fail}^s - 1)$. The latter means the transmission interval is overestimated by up to $2 \times (\pi_{fail}^s - 1)$ beacon intervals in case 1 (line 3). The upper bound is updated by using Equation 4. The value of t_{int}^s is set to the average of the two bounds. Case 3.2 covers all the remaining conditions, t_{int}^s is updated in the same way as in case 2 (lines 15–16). Case 3.3 (lines 17–18) represents the inverse of case 3.1 and 3.2, where t_{int}^s is equal to or smaller than 1 (i.e., the station is estimated to generate 1 or more than 1 packet per beacon interval). If the previous successful transmission did not cross two beacon intervals and the transmission queue is empty, t_{int}^s becomes 1, otherwise, t_{int}^s remains unchanged.

In case 4 (lines 19–26), more than 1 packet was received. In case 4.1 (lines 21–22), t_{int}^s is larger than 1 (i.e., only 1 packet was allowed to be transmitted in the previous beacon interval), the overestimated t_{int}^s is reduced by 1 beacon interval. In case 4.2 (lines 23–24), the number of received packets is higher than the estimated number of expected packets (i.e., $1/t_{int}^s$), the transmission interval is reduced by adding 1 to the inverse. Case 4.3 (lines 25–26) represents fewer packets are received than estimated and the transmission queue has no buffered packets. The transmission interval is increased by subtracting 1 from the inverse.

Finally, the next transmission time is calculated as the last successful transmission plus the newly estimated transmission interval (line 27). In essence, the algorithm is iterative, and as more information about successful and failed transmissions becomes available, the estimate of t_{int}^s will become more accurate.

Table 2: Default parameter values used in the simulation experiments

PHY parameters	Value
Frequency (Mhz)	868
TX power (dBm)	0
TX/RX gain (dB)	0
Noise Figure (dB)	6.8
Coding method	BCC
Propagation model	Outdoor, macro [6]
Error rate model	YansErrorRate
MAC parameters	Value
Duration of AIFS (us)	316
Beacon interval (ms)	100
Cross slot boundary	enabled
Rate control algorithm	constant
Size of transmission queue (packets)	10
Max/min traffic ratio between stations	20
Wi-Fi mode	MCS1, 1 Mhz
Payload size (bytes)	64
Topology radius (m)	450
Station distribution	random

The space and time complexity of E-TAROA are $O(n + k + m)$ and $O(n^2 + n \times k + m \times k)$ respectively, with n the total number of stations, k the maximum number of stations that are allowed to transmit during one beacon interval, and m the maximum number of packets received by the AP during one beacon interval. For n large enough, this can be simplified to $O(n)$ and $O(n^2)$ respectively. In 802.11ah, the maximum value of n is 8192, which provides an upper bound on the complexity of the algorithm.

4 PERFORMANCE EVALUATION

In this section we evaluate E-TAROA and compare it to the original TAROA algorithm in terms of traffic estimation accuracy, throughput and latency. The algorithms are compared in both sparse and dense deployments in three scenarios: (i) static, (ii) dynamic number of stations, and (iii) dynamic traffic patterns.

4.1 Simulation setup

The evaluations are performed using the extended version of 802.11ah ns-3 module [11], which can keep track of power consumption of each station on the basis of 802.11ah ns-3 module [12]. We target dense IoT applications, where sensors monitor the environment and report the monitoring data periodically. Each sensor has its own monitoring and transmission interval, that may change over time. We assume the transmission interval follows a uniform distribution, and the ratio between any two sensors' traffic is at most 20.

The PHY and MAC layer parameters used in the simulation are shown in Table 2. The PHY layer parameters are configured based on the low-power 802.11ah radio hardware prototype [1], with a transmission power of 0 dBm, a gain of 0 dBi, and noise figure of 6.8 dB. Considering the targeted use cases, MCS1 with 1 Mhz bandwidth (data rate 0.6 Mbps) and payload size 64 bytes are used. Stations are randomly placed around the AP in a circle of 450 m, this provides a broad coverage given the configuration, and leads

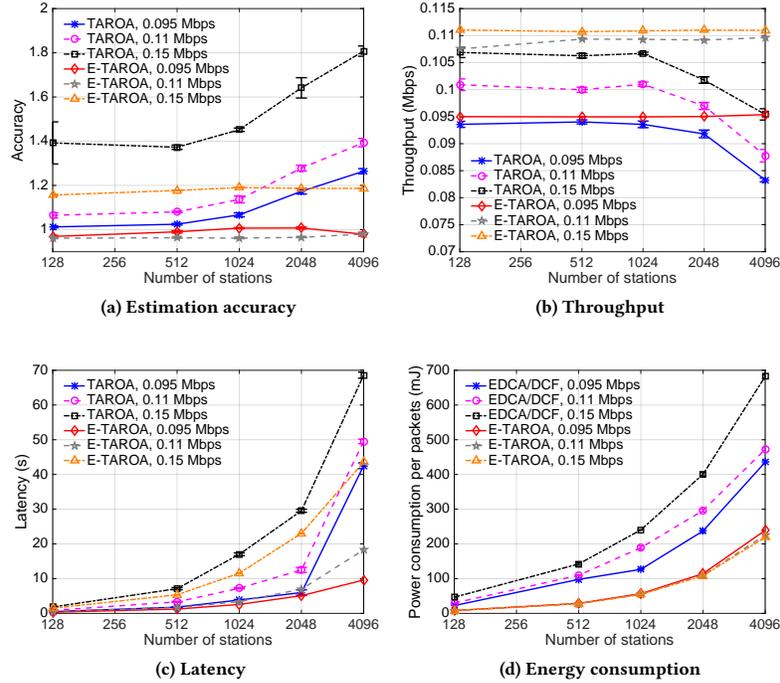


Figure 4: Performance comparison between E-TAROA, TAROA and EDCA/DCF for different static traffic loads and number of stations

to the presence of hidden nodes. The stations’ transmission queue has a capacity of 10 packets. No rate control algorithm (RCA) is used at the MAC layer.

Performance is evaluated in terms of three metrics: estimation accuracy, throughput, latency and energy consumption. The estimation accuracy represents the ratio between the estimated and the real transmission interval, averaged over all stations. As such, a value equal to 1 means no estimation error, higher than 1 means overestimation and lower than 1 means underestimation. Throughput is calculated as the average number of successfully received payload bytes by the AP per second. Latency is defined as the average time between a packet entering the transmission queue of the station and being received by the AP. Energy consumption represents the power consumed per successfully transmitted packet. Each simulation runs 900 s, all results are averaged over 10 iterations, with the variability of results over these iterations quantified using the standard deviation (SD).

4.2 Static traffic patterns

In this section we evaluate the performance of E-TAROA in a static scenario, where the number of stations in the network is fixed and they do not change their transmission interval. Three different total traffic loads $T = \{0.095, 0.11, 0.15\}$ Mbps, and 5 different number of stations $S = \{128, 512, 1024, 2048, 4096\}$ are simulated, resulting in a total of 15 types of experiments. Given the packet payload size and data rate, the maximum throughput that can be achieved is about 0.124 Mbps. Therefore, 0.095 Mbps and 0.11 Mbps represent

a low and high non-saturated traffic load, while 0.15 Mbps results in a saturated state. The traffic load and number of stations jointly determine the density of the network, with a higher traffic load and more stations resulting in a denser network.

Figure 4a depicts the accuracy of transmission interval estimation. An accuracy equal to 1 means perfect traffic interval estimation, while larger than 1 means overestimation and smaller means underestimation. It is clearly shown that E-TAROA achieves more accurate traffic estimation than TAROA in dense networks. For traffic load $T = 0.15$ Mbps, the transmission interval is overestimated by both E-TAROA and TAROA, as the traffic is overloaded and each station gets less transmission opportunities than it requires. The accuracy of TAROA changes from 1.39 ± 0.10 to 1.81 ± 0.02 from 128 to 4096 stations. However, E-TAROA only overestimates the transmission interval by at most 20 % even for 4096 stations. Under lower traffic loads $T = \{0.11, 0.095\}$ Mbps, TAROA overestimates the transmission interval, while E-TAROA underestimates it. However, the accuracy of E-TAROA is much closer to 1. Moreover, in contrast to TAROA, its accuracy almost does not degrade as the number of stations increases, resulting in better scalability.

As the final outcome of traffic estimation, the throughput, latency, and energy consumption performance are shown in Figures 4b, 4c and 4d. As paper [14] already compares TAROA and EDCA/DCF in terms of throughput and latency but does not include any power consumption results, we present the power consumption of E-TAROA and EDCA/DCF in figure 4d, and omit the other two performance of EDCA/DCF in figure 4b and 4c respectively.

The results reveal that the more accurate traffic estimation of E-TAROA also results in significantly improved throughput and latency. Moreover, it leads to higher scalability in terms of both metrics when the number of stations increases beyond 2048. For 1024 stations and less, in comparison with TAROA, E-TAROA shows most performance improvement for the high traffic load $T = 0.11$ Mbps and achieves slightly better performance than TAROA for $T = \{0.15, 0.095\}$ Mbps. With more than 2048 stations contending for channel access in the network, performance of TAROA gravely degrades, throughput drops by up to 13% and latency increases by over 10 times between 1024 and 4096 stations. The throughput of E-TAROA is more stable, with no significant degradation even up to 4096 stations. Although E-TAROA also suffers from increased latency, which is a result of the slotted nature of RAW, it is limited to a fivefold, rather than tenfold increase. Overall, compared to TAROA, up to 23% higher throughput and 77% lower latency is achieved by E-TAROA. As number of stations increase, more power is consumed by each packet in both E-TAROA and EDCA/DCF since channel contention increases, and stations has less packet to transmit but consume the same amount of power to listen to the beacon frame. However, the power consumption of E-TAROA is still much lower than EDCA/DCF, for traffic load $T = 0.15$ Mbps, EDCA/DCF consumes 46.63 and 682.13 mJ per packets for 128 and 4096 stations respectively, while only 7.85 and 219.05 mJ per packet are consumed by E-TAROA.

It is also worth mentioning that, compared to TAROA and EDCA, E-TAROA improves fairness among stations. Although CSMA (as used in EDCA/DCF) should provide each station with an equal chance to access the channel, the capture effect may allow a colliding packet with higher receive power to be received. As such, higher throughput can be obtained by stations that are closer to the AP. In contrast, by properly grouping the stations, E-TAROA and TAROA avoid this issue. The more accurate traffic estimation of E-TAROA, further improves grouping and as a consequence fairness. We evaluated fairness in terms of Jain's fairness index, using the same parameters as in Figure 4 except for evenly distributing the traffic among stations. The fairness of E-TAROA is always above 0.997, while TAROA results in at least 0.951. Finally, EDCA/DCF performs much worse, with a value between 0.395 and 0.8.

4.3 Dynamic number of stations

In this section, the ability of E-TAROA to adapt to changes in the topology (i.e., number of associated stations) over time is evaluated. The total static traffic load $T = 0.1425$ Mbps is distributed uniformly at random among 3072 stations. Simulation starts with 2048 associated stations and a traffic load of around 0.095 Mbps. A Poisson distribution is used to model the arrival and departure of stations. As the traffic of each station is fixed, station (dis)association also results in traffic variability over time.

Figure 5 compares the instantaneous throughput of E-TAROA and TAROA when the stations join and leave the network with Poisson rates 1, and 5 every second. The results suggest that E-TAROA is more resilient to network dynamics. Compared to TAROA, it achieves higher throughput and takes less time to converge. E-TAROA achieves faster convergence as it requires less packet transmissions to learn the transmission interval of stations. Although

both algorithms constantly needs to adapt the RAW configuration as the network topology changes, E-TAROA can successfully transmit all packets to the AP at Poisson rates 1, and only suffers a 8% packet loss at Poisson rate 5. In contrast, the original TAROA suffers over 12% and around 33% packet loss at Poisson rates 1 and 5 respectively. In terms of learning behavior, at Poisson rates 1, the throughput of E-TAROA and TAROA both start from simulation time 190 s, E-TAROA takes around 60 s while TAROA needs about 500 s to converge to the actual traffic pattern. Although the learning process takes around 120 s for both E-TAROA and TAROA at Poisson rate 5, the throughput of TAROA is around 27% lower. The bandwidth peak of E-TAROA that exceeds the actual generated traffic around time 300 s, is due to the stations emptying their queue after the AP has successfully learned their traffic patterns.

4.4 Dynamic Traffic

Besides stations (dis)associating over time, we also evaluate the performance when the number of active stations remains static, but instead their transmission interval changes over time. This allows us to determine the algorithm's ability to adapt its transmission interval estimate in real-time. There are 2048 stations with a total traffic load of 0.095 Mbps at the start of simulation. Every second, a random set of stations is selected to change their transmission interval, according to a Poisson distribution with rate $\lambda = 10$. For each selected station, the change in transmission interval Δ is chosen uniformly at random as a percentage between $[-20, 20]$ % or $[-50, 50]$ %. The instantaneous throughput for both E-TAROA as well as TAROA is depicted in Figure 6, which is consistent with the conclusion drawn from the results for a dynamic number of stations. Concretely, E-TAROA can quickly adapt to the traffic changes at both change steps Δ , obtaining on average the same throughput as the input traffic load. In contrast, TAROA requires much more time to adapt and the throughput drops even at rate $[-20, 20]$ %

5 CONCLUSION

This paper presents a novel traffic estimation method for RAW optimization in highly dense IEEE 802.11ah networks, and integrates it into an enhanced version of the Traffic-Adaptive RAW Optimization Algorithm (E-TAROA). This improved traffic interval estimation allows more accurate selection of station grouping parameters based on real-time dynamic traffic conditions. E-TAROA exploits the "more data" header field and cross slot boundary features, gaining more insight into the traffic conditions, accelerating the estimation of sensor station traffic patterns and avoiding false negative failed packet transmissions.

The simulation results reveal that E-TAROA achieves significantly more accurate traffic estimation than TAROA, especially for a large number of stations. The more accurate traffic estimation in turn results in highly improved throughput and latency in dense networks. Concretely, E-TAROA attains up to 23% higher throughput and 77% lower latency in dense networks with 4096 stations. Unlike the original TAROA, its performance is constant as the number of stations in the network increase. This proves its greatly improved scaling behavior in highly dense networks. Moreover, E-TAROA is more adaptable to dynamic networks in which the topology and traffic changes over time, only experiencing packet

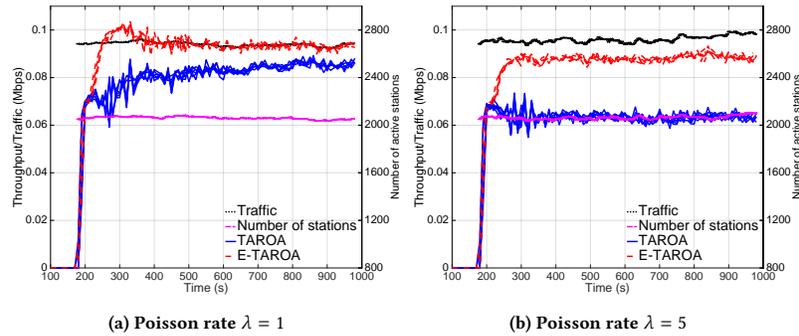


Figure 5: Instantaneous throughput comparison between E-TAROA and TAROA for stations arriving and departing over time at different Poisson rates

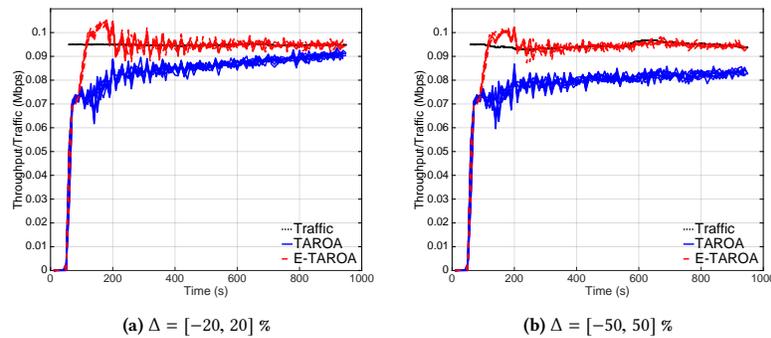


Figure 6: Instantaneous throughput comparison between E-TAROA and TAROA for stations changing their transmission interval with Poisson rate $\lambda = 10$ and different change steps Δ

loss under very high levels of station churn. Finally, by using the additional information, E-TAROA converges over 8 times as fast.

ACKNOWLEDGEMENT

Part of this research was funded by the Flemish FWO SBO S004017N IDEAL-IoT (Intelligent DENSE And Long range IoT networks) project.

REFERENCES

- [1] 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 IoT applications. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 440–441, 2016.
- [2] 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.
- [3] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso. Do LoRa low-power wide-area networks scale? In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 59–67, 2016.
- [4] 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.
- [5] A. Hazmi, B. Badihi, A. Larmo, J. Torsner, M. Valkama, et al. 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*, pages 1–6. IEEE, 2015.
- [6] A. Hazmi, J. Rinne, and M. Valkama. Feasibility study of IEEE 802.11ah radio technology for IoT and M2M use cases. In *2012 IEEE Globecom Workshops*, pages 1687–1692. IEEE, 2012.
- [7] E. Khorov, A. Krotov, and A. Lyakhov. Modelling machine type communication in IEEE 802.11ah networks. *IEEE International Conference on Communication Workshop (ICCW)*, (14):1149–1154, 2015.
- [8] 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*, E96-B(12):2976–2985, 2013.
- [9] C. W. Park, D. Hwang, and T.-J. Lee. Enhancement of IEEE 802.11ah MAC for M2M communications. *IEEE Communications Letters*, 18(7):1151–1154, 2014.
- [10] 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 (ICC)*, pages 460–466, 2014.
- [11] S. Serena, T. Le, and F. Jeroen. Ieee 802.11ah restricted access window energy consumption model for ns-3. *Workshop on NS-3 (WNS3)*, (June), 2017.
- [12] L. Tian, S. Deronne, S. Latré, and J. Famaey. Implementation and validation of an IEEE 802.11ah module for ns-3. In *Workshop on NS-3 (WNS3)*, 2016.
- [13] 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*, 2016.
- [14] L. Tian, E. Khorov, S. Latré, and J. Famaey. Real-time station grouping under dynamic traffic for IEEE 802.11ah. *Sensors*, 17(7), 2017.
- [15] 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)*, pages 581–586, 2015.
- [16] 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*, 13(11):6180–6192, 2014.