IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 16, 2020, accepted December 31, 2020, date of publication January 8, 2021, date of current version January 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3050426

Enabling Wireless Closed Loop Communication: Optimal Scheduling Over IEEE 802.11ah Networks

AMINA SEFERAGIĆ, ELI DE POORTER, AND JEROEN HOEBEKE
IDLab, Department of Information Technology, IMEC, Ghent University, B-9052 Ghent, Belgium

Corresponding author: Amina Seferagić (amina.seferagic@ugent.be)

This work was supported by the Fund for Scientific Research-Flanders through the Intelligent DEnse And Long range the IoT networks SBO Project under Grant S004017N.

ABSTRACT Industry 4.0 is being enabled by a number of new wireless technologies that emerged in the last decade, aiming to ultimately alleviate the need for wires in industrial use cases. However, wireless solutions are still neither as reliable nor as fast as their wired counterparts. Closed loop communication, a representative industrial communication scenario, requires high reliability (over 99%) and hard real-time operation, having very little tolerance for delays. Additionally, connectivity must be provided over an entire industrial side extending across hundreds of meters. IEEE 802.11ah fits this puzzle in terms of data rates and range, but it does not guarantee deterministic communication by default. Its Restricted Access Window (RAW), a new configurable medium access feature, enables flexible scheduling in dense, large-scale networks. However, the standard does not define how to configure RAW. The existing RAW configuration strategies assume uplink traffic only and are dedicated exclusively to sensors nodes. In this article, we present an integer nonlinear programming problem formulation for optimizing RAW configuration in terms of latency in closed loop communication between sensors and actuators, taking into account both uplink and downlink traffic. The model results in less than 1% of missed deadlines without any prior knowledge of the network parameters in heterogeneous time-changing networks.

INDEX TERMS IEEE 802.11ah, industrial Internet of Things (IIoT), optimization, restricted access window (RAW), Wi-Fi HaLow, wireless automation.

I. INTRODUCTION

Cabling and wiring of an industrial site (factory or plant) is expensive in many aspects, including financial cost, time needed to install racks and lay kilometers of cables, and maintenance needed due to physical degradation of wires or miswiring due to human error. The difficulty of wiring is especially accented in mobile assets (e.g. cranes, mobile robots etc.) and hard-to-reach places or places that are dangerous for humans (e.g. due to presence of radiation, gas, extreme heat etc.). Wireless communication in an industrial setting would not only mitigate the above mentioned complications, but would also introduce several logical benefits such as ''hot-swapping'' between faulty and backup module via a simple activation command, and facilitating ''plug-n-play'' automation leading to reduced downtime [1].

Although wireless technologies are gradually being introduced to industrial use cases, wireless communication is generally unwelcome in factories and plants due to its inherent unreliability. The probability of a transmitted packet to be correctly received depends on multiple factors, including the amount of power used for the transmission, the environmental impact on propagation characteristics of the channel and the collisions that may occur due to many nodes transmitting at the same time. Despite all the challenges, a wireless network needs to maintain a reasonable end-to-end delay in order to facilitate a field network in a plant. A field network consists of sensors, controller(s) and actuators communicating in a closed loop: a sensor measures the current value, reports it to the controller that makes the decision on the control value, which is then forwarded to the actuator to be executed. The cycle repeats periodically or event-based, with periods ranging from μs to ms in highly dynamic control systems (e.g. autonomous vehicle systems or motor/generator control), to minutes or hours in systems

The associate editor coordinating the review of this manuscript and approving it for publication was Nan Wu.

with slower dynamics such as tank-level or temperature control.

Considering the inherently unreliable and latency prone properties of wireless communications, proving and demonstrating the feasibility of worst case latency performance over the wireless medium is a major research challenge. More precisely, understanding what levels of scalability, latency and reliability could be guaranteed in wireless networks with high reliability are important research questions to guide the development of new radios, protocols, and applications [2], [3]. To that end, we evaluated IEEE 802.11ah in the context of low-latency time-critical Control Loops (CLs) in [4] and found that adjusting the network dynamics to that of CLs limits the jitter and meets the latency requirements, at the cost of maximum achievable throughput, network scale and energy consumption. To adjust the network dynamics to that of the fastest CL in the network, it is needed to reduce the beacon interval to at least half the value of the fastest CL's cycle time, considering that the Access Point (AP) needs to indicate the presence of downlink (DL) traffic for a node in a beacon in order for the node to wake up and receive it. Reducing the beacon interval as described is referred to as "Reduction of Beacon Interval (RBI) method" in the remainder of this article. RBI method enables timely completion of a cycle in a control loop, but also reduces the useful channel time available to nodes because of frequent beaconing, and thus maximum achievable throughput in the network. It increases energy consumption because all nodes need to wake up to receive the beacons more often. As the beacon interval in 802.11ah networks is usually set to 102.4 ms, a CL cycle time of 50 ms would require a beacon interval of 16 ms or less, resulting in 6.67 times more frequent wake-ups for receiving the beacons and taking more than 6 times more bandwidth for beaconing. To mitigate frequent beaconing while keeping timely cycles, in this article we propose a novel method for adjusting the highly configurable, Medium Access Control (MAC) feature of IEEE 802.11ah called Restricted Access Window (RAW) to CLs, while keeping large beacon intervals and estimating the future presence of DL traffic so as to indicate it in each beacon. In addition, this article reviews the spectrum regulations for IEEE 802.11ah and presents their impact to closed loop communication scenarios.

In the RBI method, dedicated RAW is assigned to every CL end node in each Beacon Interval (BI) to avoid contention with other nodes and introduce determinism (i.e. guarantee successful transmission). BI is reduced in order to over dimension the number of RAWs per loop in each cycle and enable downlink traffic indication as defined by the standard. The RAW configuration was fixed and the other non-critical nodes contended for the rest of the channel time. In this article, we keep dedicated RAWs for loops to guarantee determinism, but we do it in an adjustable RAW configuration. For fast CLs whose cycle times are smaller than the beacon interval (<100 ms), our algorithm assigns as much as needed, but as few as possible dedicated RAWs per loop per beacon interval just in time to meet the deadline for packet delivery in the loop, whereas for slower CLs it assigns RAWs accordingly infrequent. It calculates a new optimal RAW configuration every beacon interval, hence is adaptable to changes in the network (i.e. nodes joining/leaving the network) and provides as much time as possible time to other non-critical traffic that uses random access.

This article presents an Integer Nonlinear Programming (INP) problem formulation for RAW scheduling optimization in a heterogeneous network that includes both sensors and actuators. Closed loop communication requires stringent latency and timely delivery, thus we solve the problem for the optimal RAW schedule to accommodate control-loop end nodes, while minimizing channel time reserved for them so that other stations can also utilize the network. Minimizing the channel time for critical nodes implicitly minimizes their energy consumption. We use simplex algorithm to solve the INP and validate the solution through extensive experiments. Compared to WirelessHART, our solution can support up to 10 times more critical devices (40 IEEE 802.11ah control loops with 500 ms cycle time vs. 8 uplink-only WirelessHART devices with 500 ms reporting interval). Although successful, this model is computationally demanding. However, given that it should run on an access point, an external processor could compute the RAW parameters in every period and feed them to the AP without influencing the energy efficiency of the nodes. In any case, this model can serve as a baseline for designing and testing heuristic optimization algorithms for 802.11ah RAW scheduling. Finally, we study the duty cycle requirements in the unlicensed spectrum and discuss their influence on IEEE 802.11ah in this scenario, thus extending [5] and [6] which reviewed and interpreted spectrum regulations in sub-GHz domain in general.

This article is organized in eight sections. Section II summarizes related work on optimal RAW configuration in different use-cases. Section III provides an overview of the IEEE 802.11ah features relevant for this article, largely focusing on RAW. Section IV presents the challenges of applying IEEE 802.11ah to CLs and elaborates the problem. Section V gives a mathematical formulation of the INP. Section VI validates the model through extensive simulation and compares the obtained results with the state-of-the-art. Discussion on compliance with regulations in unlicensed sub-GHz spectrum in Europe is presented in Section VII. Finally, Section VIII concludes the article.

II. RELATED WORK

Wireless networks have been adapted to fit critical industrial applications, as well as developed anew in the past several years [7]. To name a few, customized Time Division Multiple Access (TDMA)-based wireless solutions such as WirelessHP [8], OFDMAwirelesscontrol [9], Real-Time-WiFi [10], w-SHARP [11] and WIA-FA [12] can achieve latency in the order of microseconds over a very short area (cca. 10 m). These solutions can be used for local control on mobile assets, production cells, robotic cells and similar concentrated areas with dedicated controllers on the spot, but not in plants where a few controllers are responsible for regulating all processes throughout the plant and are not necessarily located within several meters from all the sensors and actuators. Additionally, 5G New Radio (NR) [13] also achieves ultra low latency, similarly to the previously mentioned technologies, while extending the range to hundreds of meters, but it is managed by an external network operator. Operator based solutions are generally not ideal for industrial purposes as the dependence on the operator in case of failure increases the repair time. 802.15.4- and 802.15.1-based technologies operating in the 2.4 GHz Industrial, Scientific and Medical (ISM) band extend the coverage to the order of 100 m with direct line of sight, but also significantly increase the latency to hundreds of milliseconds, seconds or more. Large-scale plants span over several hundreds of meters and controllers are often located far away from sensors, motivating the necessity of a multi-hop network in an effort to cover a large area. However, multi-hop brings numerous disadvantages to this use case: lower throughput, latency increasing with every hop and difficulties to create and maintain a network. WirelessHART [14] and ISA100.11a [15] are two wireless technologies specifically designed for industrial use cases. Although they provide increased reliability, they do not scale well. Only 8 devices with 500 ms reporting period can be supported by a single commercial WirelessHART AP [16]. In practice, a single WirelessHART AP can support 25, 50, 80 or 100 devices at 1 s, 2 s, 4 s and 8 s reporting period (uplink) respectively. Both bidirectional (cyclic) traffic and an increment in hops doubles the latency or halves the number of nodes. ISA100.11a is even more constrained [17]. The emergence of sub-GHz technologies (e.g. LoRaWAN, Sigfox, IEEE 802.15.4g, IEEE 802.11ah) enabled increased single-hop range and robustness due to better penetration capabilities in comparison to 2.4GHz technologies. Out of the currently present sub-GHz technologies, IEEE 802.11ah has the most interesting feature set for being considered in industrial use cases. Its MAC design and higher bit rates than other long-range wireless technologies characteristic for Internet of Things (IoT) enable IEEE 802.11ah to achieve less than 2 ms latency and less than 100 ms Round-Trip Time (RTT) [4]. Thus, it can support reliable and high-throughput IoT applications (e.g. firmware updates, reliable monitoring and control) as it does not restrict the rates for DL traffic [18].

Although there are many extensive studies on IEEE 802.11ah and its RAW optimization in terms of throughput, energy consumption, latency and hidden node mitigation, a vast majority of available research considers Wireless Sensor Network (WSN) scenarios with uplink only traffic. Most of the related works that consider bidirectional communication over 802.11ah devise RAW configuration strategies with aim to reduce energy consumption or to increase throughput [19]–[22]. With the exception of our previous research [4], only one study evaluated the latency in an actuation use case (downlink) for connected lightning, at the actuator side [23]. Regardless of the different traffic patterns in different use cases, the strategies on devising the best RAW configurations for the job vary. RAW configuration consists of many parameters (RAW start time, duration, number of slots, assigned nodes etc.) that are interlocked with key performance indicators such as throughput, energy consumption, latency or scalability. Therefore, finding the optimal configuration is far from trivial. In fact, studies often propose a sub-optimal RAW configuration based on the best key performance indicators measured in a set of experiments [21], [24].

Integer programming is used to optimize the performance of IEEE 802.11ah in terms of energy efficiency [25], [26], throughput [27], [28], channel utilization [29] and hidden node mitigation [30], [31]. An integer nonlinear programming approach is employed for optimizing energy efficiency by taking into account traffic demands with even distribution of all RAW groups [25]. Another study optimizes activation/sleep scheduling to maximize network lifetime in environmental monitoring applications, while satisfying both report-accuracy and timely update requirements [26]. Throughput is optimized via non-convex integer programming optimization which avoids hidden terminals opportunistically [27]. A multi-objective optimization problem that addresses throughput maximization while minimizing unfairness across RAWs (i.e. grouping stations with similar traffic requirements) is proposed in [28]. Additionally, [28] formulated an integer problem for contention window size selection to maintain fairness among the nodes in a RAW. Furthermore, an integer programming model for load-balanced grouping problem that optimizes channel utilization of each group is presented in [29]. [30] proposes a 0/1 integer linear programming to minimize the number of hidden nodes in a group. Finally, a study [31] expands [30] by additionally considering the total traffic demand of the nodes in a group and thus optimizing their channel capacity, while minimizing the total number of RAWs if a predefined number of hidden node pairs can be tolerated. With the exception of hidden node mitigation studies, none of the above takes into account downlink traffic. The first study to optimize RAW in a scenario of emergency alert delivery in mission-critical deployments, where low latency (<10 ms) and high reliability (75%-99%) is taken into account, is published in [32]. Reference [32] proposes a mathematical model of alert delivery with RAW and optimizes RAW by minimizing consumed channel timeshare, while providing satisfactory reliability and delivery delay. This study also takes into account uplink traffic only.

III. IEEE 802.11AH OPERATION

Given that the globally used IEEE 802.11 standard, aka Wi-Fi, has been designed to provide high throughput at short distances to a few devices, it is not suitable for IoT use cases. To join the IoT playground, an amendment to the Wi-Fi family was published in May 2017 - IEEE 802.11ah, to be marketed as Wi-Fi HaLow. Ever since the draft standard IEEE 802.11ah-D1.0 appeared in October 2013 and revealed numerous novelties in comparison to previous Wi-Fi amendments, studies on this technology are regularly being published. First System on Chip (SoC) compliant with IEEE

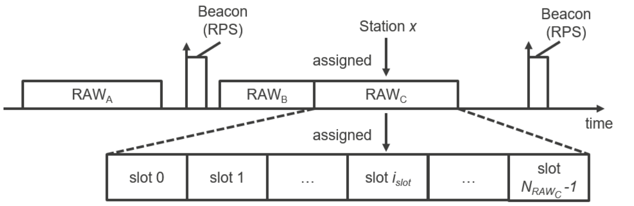


FIGURE 1. RAW configuration for all RAWs in a beacon interval is carried in the RPS element in the preceding beacon and contains (among other parameters) RAW slot duration, number of slots and assigned nodes for every RAW.

802.11ah has been released at the end of 2019 [33]. Several vendors are currently working on bringing IEEE 802.11ah to the market.

IEEE 802.11ah operates in unlicensed sub-GHz spectrum and supports 1, 2, 4, 8 and 16 MHz channel bandwidths [34]. However, depending on allocated frequency bandwidth per geographical region, not all channel bandwidths are available everywhere. In Europe, only 1MHz and 2MHz wide channels are to be used.

The standard defines two modes of operation: Traffic Indication Map (TIM) mode and non-TIM mode. TIM mode is typically used by nodes with high-bandwidth requirements or with the need for bidirectional communication, thus being the mode used in this research. TIM-nodes wake-up periodically to receive beacons broadcast by the AP, unlike non-TIM nodes which do not need to listen to beacons. The beacon interval T_{BI} denotes the time in μs between two consecutive beacons.

Each node is identified by a unique 13-bit Association ID (AID) in the range 1-8191 (AID 0 is reserved for group addressed traffic) which positions it in a 3-level hierarchy [18]. The AP uses this hierarchy to indicate the existence of pending DL traffic to a node on 3 different levels in TIMs encoded in beacons, so that nodes can quickly process TIMs in received beacons and determine if DL traffic indication exists on any of its hierarchy levels. If there indeed is a DL traffic indication for a node in the beacon, we refer to the node as *paged* by the AP. In addition to an efficient organisation of a large number of nodes and a reduction of beacon size, this hierarchy enables the nodes to sleep most of the time in order to save energy (i.e. if at the top hierarchy level a node sees no DL indication, it can skip checking the lower hierarchy levels and sleep longer) [18].

To further reduce the energy consumption by increasing the quantization of time for sleeping, as well as to reduce collisions and interference, consequentially increasing the throughput, the standard introduces the RAW feature (cf. Fig. 1) [18]. RAW reserves a specific slotted time window for a specific group of nodes. Slots inside a RAW are equally long and nodes assigned to a RAW are evenly split across the slots using round robin assignment. A node employs a random access method, Enhanced Distributed Channel Access (EDCA)/Distributed Coordination Function (DCF), to access the medium in its slot and will not contend for the medium in any other RAW slot during that RAW. Other nodes that are not assigned to the RAW are not allowed to contend for the medium during that RAW. Nodes assigned to a RAW may or may not cross the slot boundary (e.g. to finish an ongoing frame exchange sequence) depending on the RAW configuration. RAW configuration for every RAW in a beacon interval is encoded in a RAW Parameter Set (RPS) element broadcast to the nodes in a beacon at the beginning of the beacon interval. RAW duration is defined by the Slot Duration Count C, Slot Format and the Number of Slots N_{RAW}. As per the IEEE 802.11ah standard [34], the duration of the RAW slot T_{slot} is defined by

$$T_{slot} = 500\mu s + C \cdot 120\mu s. \quad (1)$$

The duration of the entire RAW is thus $N_{RAW} \cdot T_{slot}$. The minimal and maximal slot duration are determined by the non-negative integer C, which is either 11 or 8 bits long depending on whether the Slot Format is set to 1 or 0, respectively. Apart from mentioned parameters that define RAW, it is important to mention Start AID and End AID of the nodes assigned to a RAW. This means that only nodes with sequential AIDs can be assigned to the same RAW. To devise a meaningful grouping strategy, AID assignment must also be devised accordingly.

IEEE 802.11ah introduced the Target Wake Time (TWT) feature which enables a pair of stations to negotiate the future time(s) to wake up and exchange frames, hence omitting the need to receive beacons, consequentially conserving the energy. TWT is intended to provide a low-consumption mode for stations with low traffic requirements and periodic data transmissions. A station that initiates a TWT agreement (i.e. TWT requesting station) is assigned a specific times to wake and exchange frames with TWT responding station. Responding station is the one that devises the schedule and delivers it to the TWT requesting station. Two types of TWT agreement are defined: explicit and implicit TWT. When explicit TWT is employed, TWT requesting station receives the next TWT value in the response from the responding station during the frame exchange. Implicit TWT provides a fixed period information and requesting station calculates the next TWT by adding the fixed period to the current TWT value. Both stations conserve energy by omitting the need to receive beacons.

A TWT requesting station may request the TWT responding station to protect the set of TWTs by allocating RAW(s) that restrict access to the medium during the TWT service period(s) for that(those) TWTs. This is, however, possible only if TWT responding station is an AP. Otherwise, a non-AP responding station could protect the TWTs with Network Allocation Vector (NAV)-setting frame exchanges or some other protection mechanisms. It is important to note that TWT requesting station is not granted any special medium access privileges by default, nor is there any guarantee that the TWT responding station assigned the TWT service period to only one TWT requesting station [34].

IV. CHALLENGES AND APPROACH

We consider a generic industrial site (e.g. plant, factory) hosting a large number of sensors, a (smaller) number of actuators and one or a few Programmable Logic Controllers (PLCs). Two distinct sensor operations are considered: regulatory operation in closed loop, and purely indicative operation for supervisory applications (e.g. progress monitoring, collection of trends, environmental conditions monitoring). Precisely, the modelled network consists of:

- a master node (i.e. controller), which is usually wired to the AP in practice [35],
- n slave nodes (i.e. critical sensors/actuators) which take part in wireless closed loop control, ergo communicate the measurements to the master node over the AP in uplink and receive the actuation commands from the master node in downlink,
- $N - n$ non-critical sensor nodes that do not take part in wireless control loops, but serve indicative (uplink-only) applications.

Closed loop communication of slave nodes is organized in a cyclic fashion as follows. Slaves send measurements of the controlled processes to the AP, over 802.11ah wireless network. The AP forwards the packets to the master node over Ethernet. The master node computes the control signal in time t_{proc} and sends it back to the AP over a wire. The AP then passes the control packet to the slave node over 802.11ah. In a cycle, delays occur in (1) both sensing and actuation since neither is immediate in real world, (2) computing the control values (processing time t_{proc}) and (3) communication, the most significant component of the cumulative delays. To maintain the stability of the system, the k^{th} actuation must complete before the $(k+1)^{th}$ measurement is scheduled to be sent, taking into account aforementioned delays. In other words, the deadline for completing the k^{th} cycle is $k \cdot T_h^{cyc}$, $k \in \mathbb{N}$, where T_h^{cyc} is the cycle time of slave node h (i.e. the time between two consecutive measurements).

In addition to communication range and reliability of packet delivery, one of the major issues in such scenarios is meeting the deadlines with respect to the cycle times. IEEE 802.11ah provides sufficient range to cover an industrial site. Using 2 MHz channels, it can cover diameters ranging from 200 m with Modulation and Coding Scheme (MCS) index 8, up to 1 km with MCS index 0 [4], [36]. To avoid contending for channel access with other nodes, thus guaranteeing the channel access opportunity to a slave node and maximizing its reliability under the given channel conditions, we opted to use a dedicated RAW per slave node. This protects the medium exclusively for slave nodes. It prevents them from backing off inside their assigned RAWs and risking to miss their Transmission Opportunities (TXOPs), hence introducing determinism to the slaves at the cost of reducing the available channel time to the $N - n$ non-critical sensor nodes. Non-critical sensors report their measurements to the supervisory systems or data bases periodically or event based,

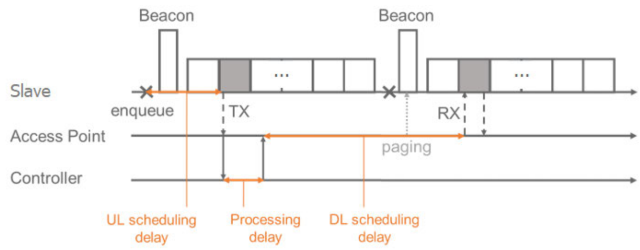


FIGURE 2. Different components of IEEE 802.11ah communication delays in bidirectional communication.

and they contend for the medium between critical RAWs as they can tolerate some delays and losses.

TWT mechanism may also be suitable for precise scheduling of transmissions, but not for the reliable delivery before deadlines. Although TWT might help enabling control loops without frequent beaconing in principle, TWT does not provide any medium access privileges to the stations. Therefore, RAW is necessary for providing reliability guarantees by avoiding any possibility of contention at the targeted times, especially considering high traffic requirements of control loops with short cycle times. TWT may be used in combination with RAW for slow control loops with low traffic demands, but fast control loops with cycle times below 100 ms are not a suitable candidate for TWT given the overheads and negotiation delays. Without TWT, the AP must first indicate the presence of DL traffic in a beacon to ensure the slave will be awake to receive the DL data. As illustrated in Fig. 2, this can introduce substantial delay in downlink because the slaves need to wait for the next beacon to be notified that they have pending DL data, and then wait for their RAW slot after the next beacon to retrieve their DL data. This makes the CLs with cycle times shorter than the beacon interval infeasible with IEEE 802.11ah without TWT. To mitigate this issue, we estimate the future availability of the DL data for each following beacon interval and page the corresponding nodes in advance, as detailed below.

Every slave node h needs two RAWs per its cycle time T_h^{cyc}, one to transmit an uplink (UL) packet and the other to receive a DL response. Note that the RAW configuration for each beacon interval is announced at the preceding beacon. Furthermore, any cycle time T_h^{cyc} can be expressed as a number of units of the beacon interval T_{BI} as $T_h^{cyc} = x \cdot T_{BI}$, where $x \in \mathbb{R}^+$. For the sake of simplicity, let us consider $x \in \{p/q \mid GCD(p,q) = 1 \wedge p, q \in \mathbb{N}\}$ only, where GCD stands for the greatest common divisor. The number of UL packets to be transmitted per slave node per beacon interval is constant only for $x = \{1/q \mid T_{BI} \mod q = 0 \wedge q \in \mathbb{N}\}$. For all other values of x, the number of UL packets per slave node per beacon interval changes from one beacon interval to another, but the pattern repeats periodically every p beacon intervals. Although, theoretically, one could devise a list of RPS's such that all slaves are assigned to as many RAWs as they need in each beacon interval, the length of such list would be the least common multiple of all values of p of all slave nodes. Furthermore, any approximation of T_h^{cyc} to $\frac{p}{q} \cdot T_{BI}$ would result in imprecise scheduling. Even if the imprecise scheduling and the length of the RPS list do not pose much of an issue, which is unrealistic to expect, constructing such RPS elements is far from trivial considering that one has to take into account all cycle times of all slaves and assign non-overlapping RAWs to them. For these reasons, we developed a INP model which solves the RAW scheduling problem for a set of slave nodes with different and variable cycle times $T_h^{cyc} \in \mathbb{R}^+$, while maximizing the channel time available for other non-critical sensor nodes. Using INP solution, the AP determines the RPS for the upcoming beacon interval, taking into account all cycle times of all slave nodes and assigning non-overlapping RAWs to them.

Minimizing the channel time reserved for slave nodes implicitly minimizes their energy consumption. Critical nodes in a loop operate in an orderly cyclic fashion and must make a fixed number of successful transmissions (TXs) and receptions (RXs) in an observed period. Energy will be dissipated if retransmissions occur, both in UL and DL, where UL retransmissions dissipate more energy (TX state) than the idle (i.e. passive awake) state when expected DL transmission fails to arrive. Dedicated RAWs erase the possibility of collisions within a network, hence ensuring the successful transmissions (at least from MAC layer's perspective). Considering that INP minimizes the duration of RAWs, INP indirectly minimizes the energy consumption of slave nodes.

V. INP: INTEGER NONLINEAR PROGRAMMING MODEL

The goal is to minimize channel time used by slave nodes in the network, provided that every slave gets both a timely transmission opportunity (UL RAW) and a reception opportunity (DL RAW) within its cycle, if possible. A RAW is completely defined with the number of slots, slot start time, slot duration, and AIDs of the nodes assigned to it. INP assigns slave nodes to single-slot RAWs of minimal duration that satisfy the scheduling criteria. The rest of the channel time (i.e. channel time not reserved for slave nodes) can be utilized for non-critical traffic. INP calculates the RAW duration of all RAWs, both for slaves and other nodes. Other nodes are assigned to all the RAWs that are not assigned to the slaves.

INP is solved at the AP at the beginning of every beacon interval, before the AP broadcasts a beacon, given that the AP must define and broadcast RPS with the beacon. To calculate the optimal RPS, the AP must collect measurements from the network in order to adjust to it in every beacon interval. The measurements that the AP collects are summarized in Section V-A. The reasoning and the mathematical formulation of the INP model is presented in Section V-B.

A. INPUT TO THE INP

Table 1 summarizes the values that the AP collects (measures or calculates from the measurements) in every beacon interval in order to determine the optimal RPS to be used in the next beacon interval. Input variables that are characteristic for each particular slave node $h \in \{0, 1, 2, \ldots, n-1\}$ are denoted with the index h and the INP considers n values of each such variable (one for every slave node).

TABLE 1. Input to the INP.

Name	Count	Type	Description
t	1	Integer	Time when INP is solved in μs (equals start time of current BI)
m	1	Integer	Max. number of RAWs
T_{BI}	1	Integer	Beacon interval in μs
T_{CH}^{max}	1	Integer	Max. channel time in μs $T_{CH}^{max} = T_{BI} - 2040\mu s$
p_h	n	Binary	Paged node indication of node h
ϵ_h	n	Binary	Indication if the last packet of node h in the BI *must* be delivered in this BI
y_h	n	Binary	Outstanding packets indication of node h
T_h^{cyc}	n	Integer	Cycle time of node h in μs
t_h^s	n	Integer	Timestamp of the last packet of node h
t_h^{sprev}	n	Integer	Timestamp of 2^{nd}-to-last packet of node h
l_{bitmap}	1	Integer	Page bitmap size
n_{paged}^{TIM}	1	Integer	Num. of paged TIMs
$n_{paged}^{subblocks}$	1	Integer	Num. of paged subblocks
N_h^{UL}	n	Integer	Num. of UL packets that node h has scheduled in the next BI
t_{TX}^P	1	Integer	packet transmission duration in μs
t_{proc}	1	Integer	Processing time at the controller in μs
$t_h^{RAW,prev}$	n	Integer	Start time of the last RAW node h was assigned to in the prev. BI
t_{BI}^{TX}	1	Integer	Time reserved for beacon transmission in the next iteration

The AP collects the measurements and solves the INP at the beginning of each beacon interval before broadcasting a beacon, at time t (cf. Fig. 3). Every slave node adds a timestamp of the measurement to each UL packet. The AP inspects the packets and extracts the timestamps, in order to precisely calculate the cycle time of each loop as $T_h^{cyc} = t_h^s - t_h^{Sprev}$. INP takes into account only those nodes that have initialized all the input parameters listed in Table 1. Hence, the transition process to the optimal scheduling of the entire network takes $2 \cdot \max\{T_h^{cyc}\}$ due to the need to populate t_h^s and t_h^{Sprev}. However, the AP starts using INP as soon as the measurements become available for any slave node, and updates the list of ready nodes in every beacon interval until all nodes are accounted for. Parameters l_{bitmap}, n_{paged}^{TIM} and $n_{paged}^{subblocks}$ impact the size of the beacon to be broadcast and can be read from the TIM. The transmission time of a beacon depends on its size and deducts that time from the useful channel time that nodes can use from the beacon interval. Thus, the maximal possible useful channel time equals $T_{CH}^{max} = T_{BI} - 2040\mu s$ where $2040\mu s$ is the time needed to broadcast the minimal-sized beacon (65 bytes) with both empty TIM and empty RPS. Both TIM and RPS add overhead to the beacon and reduce the useful channel time for the nodes.

p_h indicates if node h is paged. A node is paged in the beacon when there is a pending packet at the AP waiting to be delivered to the node in the following beacon interval. We consider a somewhat loose definition of the term

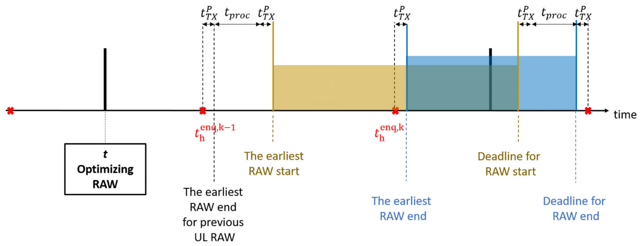


FIGURE 3. The AP can delay the RAWs for the k^{th} packet for the BI after the one being calculated at time t because the deadline for RAW start spans beyond the current BI.

"paged", also including packets that are not available at the AP yet due to the processing delay, but will be during the upcoming beacon interval. The processing time at the controller, t_{proc}, is the time the master node needs to calculate the control value from the received measurement. Normally, the AP should only indicate a pending DL packet to the slave node once it has it available in its queue. However, assuming that t_{proc} is sufficiently smaller than the cycle time, the AP can conclude it "should" also page a slave as soon as it receives its UL packet, in spite of not having the DL packet from the master at the moment of broadcasting the beacon, but knowing that the response from the master node will arrive in the upcoming BI. Therefore, we consider p_h to become 1 as soon as the AP receives an UL packet from the slave h, and to reset upon downlink forwarding of the packet. If a slave is paged, it needs to receive pending DL packets before its next scheduled transmission.

$$p_h = \begin{cases} 1, & \text{if the node } h \text{ is paged,} \\ 0, & \text{otherwise.} \end{cases} \qquad \forall h \in \{0, \ldots, n-1\} \tag{2}$$

A node might have outstanding packets in uplink, i.e. the packets already enqueued at the node waiting to be transmitted but not yet transmitted in the preceding beacon interval as those nodes did not have a TXOP after they scheduled the packets. The indicator of outstanding UL packets is defined as

$$y_h = \begin{cases} 1, & \text{if } \left\lfloor \dfrac{t - t_h^s}{T_h^{cyc}} \right\rfloor > 0, \\ 0, & \text{otherwise.} \end{cases} \qquad \forall h \in \{0, \ldots, n-1\}. \tag{3}$$

We use the term *following BI* for the BI that immediately follows the beacon where INP is solved at time t. Hence, following BI spans from time t to $t + T_{BI}$. INP solution (RAW configuration) is, thus, applied to the *following* BI. We use the term *next BI* for the beacon interval that immediately follows the following BI, i.e. from time $t + T_{BI}$ to $t + 2T_{BI}$.

The deadline for completing the cycle of the slave's last packet in the following BI for which the INP optimizes RAW configuration is in the next BI, as illustrated in Fig. 3. Thus, the last cycle could be completed in the following BI; it could stretch over both the following and the next BI (slave would be paged in the next iteration) or it could complete in the next BI if there is enough time available. Let ϵ_h indicate if there is enough time in the next BI to deliver an outstanding UL packet (as well as a DL response to it). As Fig. 3 shows, there is enough time in the next BI if the deadline for RAW start for the outstanding uplink packet is in that BI, and not before the corresponding beacon transmission is finished. ϵ_h figures in constrains that limit the number of RAWs assigned to slave nodes and equals

$$\epsilon_h = \begin{cases} 1, & \text{for expression } A_h, \\ 0, & \text{otherwise.} \end{cases}$$

$$A_h \equiv t_h^s + T_h^{cyc}(N_h^{UL} + y_h + 1) - 2t_{TX}^P - t_{proc} \leq t + T_{BI} + t_{BI}^{TX} \qquad \forall h \in \{0, \ldots, n-1\}. \tag{4}$$

In simpler terms, ϵ_h equals 0 if there is enough time before the deadline to transmit the packet in the next BI. ϵ_h equals 1 if the packet must be transmitted in the following BI for which the AP is about to broadcast the beacon, in order to meet the deadline.

The number of transmissions to be scheduled in the following beacon interval by each slave node h depends on the node's T_h^{cyc} and equals

$$N_h^{UL} = \left\lfloor \frac{t + T_{BI} - t_h^s}{T_h^{cyc}} \right\rfloor \qquad \forall h \in \{0, \ldots, n-1\}. \tag{5}$$

B. MATHEMATICAL FORMULATION

A network consists of n slave nodes, were each node h schedules a transmission periodically every T_h^{cyc}. The AP knows all the values listed in Table 1 $\forall h \in \{0, \ldots, n-1\}$.

n slave nodes are assigned to up to m single-slotted RAWs, where each RAW has duration of c_i, $i = \{0, \ldots, m-1\}$. The RAWs are sequential in time, and the ones that do not accommodate slave nodes can accommodate other nodes that do not take part in considered closed loop control application. Some slots may not be needed and their duration in the INP solution will be zero. Decision variables of the proposed INP are listed in Table 2.

The goal is to optimize the scheduling, i.e. optimally distribute the RAWs to give a TX and a RX opportunity to each slave node whenever it needs one, if possible, while maximizing the channel time for other traffic. The constrains (8) through (37) enforce such distribution of RAWs that ensures real-time operation (i.e. timely completion of all cycles of all nodes), while the following objective function minimizes the channel time for slave nodes

$$\min \sum_{i=0}^{m-1} \sum_{h=0}^{n-1} w_{ih} c_i. \tag{6}$$

Let w_{ih} indicate if RAW i is assigned to slave node h

$$w_{ih} = \begin{cases} 1, & \text{if RAW } i \text{ accommodates slave node } h, \\ 0, & \text{otherwise.} \end{cases}$$

TABLE 2. Decision variables in the INP.

Name	Range	Count	Type	Value limits Min	Value limits Max	Description
c_i	$0 \le i < m$	m	Integer	0	T_{CH}^{max}	RAW duration in μs
w_{ih}	$0 \le i < m$ $0 \le h < n$	mn	Binary	0	1	RAW assignment
u_{ih}	$0 \le i < m$ $0 \le h < n$	mn	Binary	0	1	Uplink slot indication
d_{ih}	$0 \le i < m$ $0 \le h < n$	mn	Binary	0	1	Downlink slot indication
m_{eff}		1	Integer	0	m	Num. of RAWs that have duration > 0
T_{CH}		1	Integer	0	T_{CH}^{max}	Available channel time in μs
a		1	Integer	87	$93 + 4m$	Num. of symbols in a beacon frame
t_i^{start}	$0 \le i < m$	m	Integer	$t + T_{BI} - T_{CH}$	$t + T_{BI}$	Start time of slot i in μs
t_i^{end}	$0 \le i < m$	m	Integer	$t + T_{BI} - T_{CH}$	$t + T_{BI}$	End time of slot i in μs
f_{ih}	$0 \le i < m$ $0 \le h < n$	mn	Integer	0	$m/2$	$f_{ih} = \left\lfloor \frac{\sum_{k=0}^{i-1} w_{kh}}{2} \right\rfloor$
s_{ih}	$0 \le i < m$ $0 \le h < n$	mn	Integer	0	$m/2$	$s_{ih} = \left\lceil \frac{\sum_{k=0}^{i-1} w_{kh}}{2} \right\rceil$
r_i	$0 \le i < m$	m	Binary	0	1	$c_i > 0$
v_{ijh}	$0 \le h < n$ $0 \le i < m-1$ $i+1 \le j < m$	$\frac{nm(m-1)}{2}$	Binary	0	1	$v_{ijh} = u_{ih} d_{jh}$

$$\forall i \in \{0, \dots, m-1\}, \ \forall h \in \{0, \dots, n-1\}. \quad (7)$$

We assume that there are $N \ge n$ nodes in the network in total, where $AID = \{1, 2, \dots, n\}$ are assigned to slave nodes, whereas $AID = \{n+1, \dots, N\}$ are assigned to the other nodes. Hence, $w_{ih} = 1$ as defined in (7) can be understood as "slave node with $AID = h + 1$ is assigned to RAW indexed i". INP solution gives optimal RAWs for slave nodes only, and the rest (if any) will be available to $N - n$ other nodes. Although all RAWs $i \in \{0, \dots, m-1\}$ for which $w_{ih} = 0 \ \forall h \in \{0, \dots, n-1\}$ will be assigned to other nodes after the computation of the optimal solution, they will be referred to as *empty RAWs* in the remainder of this article (alluding "empty of slave nodes", as 0 slave nodes are assigned to them).

Every RAW must accommodate one or no slave nodes, as expressed in (8).

$$\sum_{h=0}^{n-1} w_{ih} \le 1 \quad \forall i \in \{0, \dots, m-1\}. \quad (8)$$

Given that the number of all decision variables must be fixed, the maximum number of RAWs m has to be predefined. However, some of the m RAWs will neither be used for slaves nor for other nodes. Thus, some RAWs might truly be empty (not just empty of slave nodes, but also empty of other nodes). Because of that, we must allow those RAWs to have 0 duration, although used RAWs must have non-zero duration. Since this INP assigns RAWs only to the slave nodes, and the rest of RAWs will be assigned once the optimal slaves' RAWs are defined via INP solution, all RAWs to which a slave node is assigned must have non-zero duration, whereas the rest may (or may not) have 0 duration, as defined in (9).

$$t_{TX}^{p} \sum_{h=0}^{n-1} w_{ih} \le c_i \le T_{CH}^{max} \quad \forall i \in \{0, \dots, m-1\}. \quad (9)$$

Note that for non-empty RAWs (i.e. the ones for which $\sum_{h=0}^{n-1} w_{ih} = 1$) we constrain the minimal RAW duration to t_{TX}^{p} to ensure that a packet can be transmitted in that RAW without crossing its boundary and possibly interfering with other nodes.

Slaves can have various cycle times, both smaller and larger than the BI. The smaller ones need one or more RAWs within a BI, whereas the larger might need no RAWs at all in some BIs. The following constraint assigns a number of

RAWs to each slave node

$$\sum_{i=0}^{m-1} w_{ih} \geq 0 \quad \forall h \in \{0, \ldots, n-1\}. \tag{10}$$

Out of m RAWs, the effective number of RAWs that actually will be encoded in the beacon is the number of RAWs with non-zero duration. That includes both non-empty RAWs (assigned to slave nodes by INP) and *empty* RAWs with non-zero duration (that will be assigned to other nodes before broadcasting the beacon). Thus, the effective number of RAWs $m_{eff} \leq m$ with a non-zero duration $c_i > 0$ is

$$m_{eff} = \sum_{i=0}^{m-1} r_i, \tag{11}$$

where $r_i = \left\lceil \frac{c_i}{c_i+1} \right\rceil$ indicates if i^{th} RAW slot exists or not. More specifically, r_i equals 1 if $c_i > 0$, and equals 0 if $c_i = 0$. To avoid the ceiling function, $r_i \in \{0, 1\}$ is defined as

$$\frac{c_i}{c_i+1} \leq r_i \leq \frac{c_i}{c_i+1} + 0.99 \quad \forall i \in \{0, \ldots, m-1\}. \tag{12}$$

Out of m_{eff} RAWs, $\sum_{h=0}^{n-1} \sum_{i=0}^{m-1} w_{ih}$ RAWs are assigned to slave nodes. Every RAW adds up to the overhead of the beacon, thus increasing the beacon transmission time and thereby reducing the remaining channel time available to the nodes. Beacon transmission time in μs equals to $240 + 40a$, where $240\mu s$ denotes the preamble and header duration, $40\mu s$ the symbol duration and a the number of symbols in a beacon frame as a function of m_{eff} [34]:

$$a \geq \frac{14+8 \cdot \left(65 + l_{bitmap} + n_{paged}^{TIM}(63 + n_{paged}^{subblocks}) + 6m_{eff}\right)}{12}, \tag{13}$$

$$a \leq \frac{14+8 \cdot \left(65 + l_{bitmap} + n_{paged}^{TIM}(63 + n_{paged}^{subblocks}) + 6m_{eff}\right)}{12} + 0.99. \tag{14}$$

Useful channel time (in μs) available to nodes is

$$T_{CH} = T_{BI} - (240 + 40a). \tag{15}$$

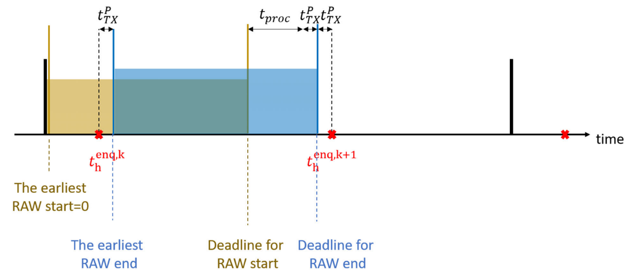
RAWs can only use the useful channel time, as in

$$\sum_{i=0}^{m-1} c_i \leq T_{CH}. \tag{16}$$

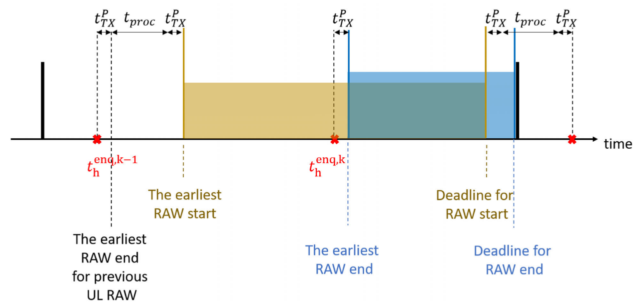
Each slave node is assigned at most 1 RAW for transmitting each UL packet (including the outstanding ones) and 1 RAW for receiving each DL packet:

$$\sum_{i=0}^{m-1} w_{ih} \leq p_h + 2y_h + 2N_h^{UL} \quad \forall h \in \{0, \ldots, n-1\}. \tag{17}$$

The minimal number of RAWs allowed in the solution depends on the number of expected packets per slave in the following BI, as well as on the feasibility of delaying slaves' RAWs for the next BI:



(a) Uplink RAW for the first packet in the beacon interval, or an outstanding packet (if any), must start anywhere between the beginning of the beacon interval and $t_{proc} + 2t_{TX}^P$ before the next enqueue time.



(b) Uplink RAW for every next packet in the beacon interval must start anywhere between $t_{proc}+2t_{TX}^P$ after the previous enqueue time and $t_{proc}+2t_{TX}^P$ before the next enqueue time (or end of BI). The RAW must end anywhere between t_{TX}^P after the enqueue time of the packet to be transmitted in that RAW and t_{TX}^P before the next enqueue time (or end of BI).

FIGURE 4. Uplink RAW scheduling for (a) the first packet in a BI and (b) the last packet in a BI.

$$\sum_{i=0}^{m-1} w_{ih} \geq 2\left(N_h^{UL} + y_h - \left\lceil \frac{N_h^{UL} + y_h + \epsilon_h}{N_h^{UL} + y_h + \epsilon_h + 1} \right\rceil + \epsilon_h\right) + p_h \quad \forall h \in \{0, \ldots, n-1\}. \tag{18}$$

Constrains on RAW start and end time are:

$$t_i^{start} = t + T_{BI} - T_{CH} + \sum_{j=0}^{i-1} c_j \quad \forall i \in \{0, \ldots, m-1\}, \tag{19}$$

$$t_i^{end} = t_i^{start} + c_i \quad \forall i \in \{0, \ldots, m-1\}. \tag{20}$$

Consider UL packets to be scheduled in the following BI (or an outstanding packet) as illustrated in Fig. 4. Slave node h's RAW slot for transmitting k-th packet must end the earliest t_{TX} after $t_h^{enq,k}$, to allow the minimal time for a single packet TX after the packet is enqueued. It must end the latest before $t_h^{enq,k+1} - t_{TX}$ to leave the minimal time for a single packet reception (t_{TX}) to receive the response (in DL RAW). Finally, it must start no later than $t_h^{enq,k+1} - 2t_{TX} - t_{proc}$, to have enough time to transmit the packet, wait for the processing, and ensure there is enough time to receive the response before the next packet is ready to send.

To mathematically describe slot start and slot end constrains (for UL slots), INP needs to determine which slots are uplink and which are downlink, for each particular node. More precisely, it needs to know the index i of the first, second, etc. up to the N_h^{UL}-th UL slot assigned to node h. If RAWs can be optimally distributed to all slave nodes, in a way they provide TXOP in time for every transmission and reception, there will never be more than 1 outstanding packet per node. Similarly, there will never be more than 1 downlink packet pending to be delivered. Also, outstanding UL and pending DL packets cannot occur simultaneously for the same node. Thus, when there is an outstanding packet or when there are no outstanding uplink nor pending downlink packets, first RAW assigned to the slave node will be used for UL. Otherwise, when there is a pending DL packet ($p_h = 1$), first RAW assigned to the node must be DL. Therefore, INP can determine which slot is UL and which is DL for each node and each slot by examining the paged status of a node and taking into account subsequent (even/odd) slot assignments. Note that p_h becomes 1 as soon as AP receives an uplink packet, in spite of not receiving the response to it from the controller before t_{proc} expires. Constrains (21) and (22) determine if slot i (provided that it is assigned to node h) is used for uplink or downlink transmissions respectively.

$$u_{ih} = w_{ih}\Big(\big(1-p_h\big)\big(1-s_{ih}+f_{ih}\big)+p_h\big(s_{ih}-f_{ih}\big)\Big) \quad (21)$$

$$d_{ih} = w_{ih}\Big(\big(1-p_h\big)\big(s_{ih}-f_{ih}\big)+p_h\big(1-s_{ih}+f_{ih}\big)\Big) \quad i \in \{0,\dots,m-1\},\ h \in \{0,\dots,n-1\}, \quad (22)$$

where $f_{ih} = \left\lfloor \frac{\sum_{k=0}^{i-1} w_{kh}}{2} \right\rfloor$ and $s_{ih} = \left\lceil \frac{\sum_{k=0}^{i-1} w_{kh}}{2} \right\rceil$ are defined by the following constrains, mitigating the ceiling and floor functions:

$$\frac{\sum_{k=0}^{i-1} w_{kh}}{2} - 0.5 \le f_{ih} \le \frac{\sum_{k=0}^{i-1} w_{kh}}{2} \quad i \in \{0,\dots,m-1\},\ h \in \{0,\dots,n-1\}, \quad (23)$$

$$\frac{\sum_{k=0}^{i-1} w_{kh}}{2} \le s_{ih} \le \frac{\sum_{k=0}^{i-1} w_{kh}}{2} + 0.5 \quad i \in \{0,\dots,m-1\},\ h \in \{0,\dots,n-1\}. \quad (24)$$

When the first slot assigned to node h in the BI is UL, then $p_h = 0$ and the term multiplied by p_h will be neutral in the (21), thus only the term multiplied by $1 - p_h$ will influence the result, provided that the observed slot i is indeed assigned to node h. DL RAWs are determined vice versa.

Timing constrains for UL RAWs, as illustrated in Fig. 3 and Fig. 4 are:

$$t_i^{start} \le t_i^{end} \quad \forall i \in \{0,\dots,m-1\}, \quad (25)$$

$$t_i^{start} \ge \sum_{h=0}^{n-1} u_{ih}(t_h^s + T_h^{cyc} f_{ih} + 2t_{TX}^P + t_{proc}) \quad \forall i \in \{0,\dots,m-1\}, \quad (26)$$

$$t_i^{start} u_{ih} \le t_h^s + T_h^{cyc}(f_{ih}+2) - 2t_{TX}^P - t_{proc} \quad \forall i \in \{0,\dots,m-1\},\ \forall h \in \{0,\dots,n-1\}, \quad (27)$$

$$t_i^{end} \ge t + T_{BI} - T_{CH} \quad \forall i \in \{0,\dots,m-1\}, \quad (28)$$

$$t_i^{end} \le t + T_{BI} \quad \forall i \in \{0,\dots,m-1\}, \quad (29)$$

$$t_i^{end} \ge \sum_{h=0}^{n-1} u_{ih}\Big(t_h^s + T_h^{cyc}\big(f_{ih}+1\big) + t_{TX}^P\Big) \quad \forall i \in \{0,\dots,m-1\}, \quad (30)$$

$$u_{ih} t_i^{end} \le t_h^s + T_h^{cyc}\big(f_{ih}+2\big) - t_{TX}^P \quad \forall i \in \{0,1,\dots,m-1\},\ \forall h \in \{0,\dots,n-1\}. \quad (31)$$

Slave node h's RAW slot for receiving a response to k-th packet should start no later than $t_h^{enq,k+1} - t_{TX}$ and no sooner than the end of preceding UL RAW. It should end no sooner than $t_h^{enq,k} + 2t_{TX}^P + t_{proc}$ and no later than $t_h^{enq,k+1}$, as illustrated in Fig. 5.

Following the same principles as for uplink slots and considering Fig. 5, timing constrains on DL slots are:

$$t_i^{start} \ge \sum_{h=0}^{n-1} d_{ih}\Big(t_h^s + T_h^{cyc} s_{ih} + t_{TX}^P\Big) \quad \forall i \in \{0,\dots,m-1\}, \quad (32)$$

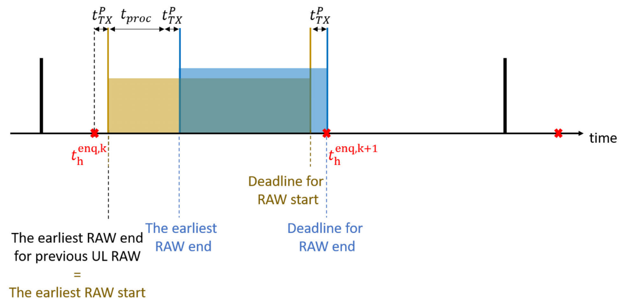
$$d_{ih} t_i^{start} \le t_h^s + T_h^{cyc}\big(s_{ih}+1\big) - t_{TX}^P \quad \forall i \in \{0,\dots,m-1\},\ \forall h \in \{0,\dots,n-1\}, \quad (33)$$

$$t_i^{end} \ge \sum_{h=0}^{n-1} d_{ih}\Big(t_h^s + T_h^{cyc} s_{ih} + 2t_{TX}^P + t_{proc}\Big) \quad \forall i \in \{0,\dots,m-1\}, \quad (34)$$

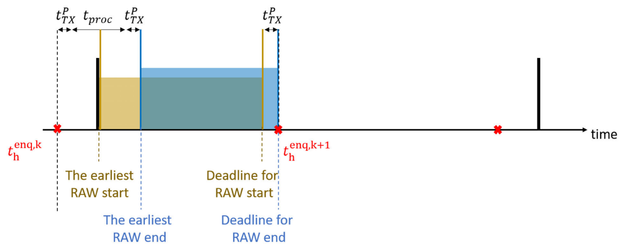
$$d_{ih} t_i^{end} \le t_h^s + T_h^{cyc}\big(s_{ih}+1\big) \quad \forall i \in \{0,\dots,m-1\},\ \forall h \in \{0,\dots,n-1\}. \quad (35)$$

Constraint (36) denotes that a DL RAW always occurs after its corresponding UL RAW. Constraint (34) reflects the Fig. 5b. However, the earliest end time for a DL RAW depicted in Fig. 5b only stands if the corresponding UL packet was delivered in the previous BI as soon as the measurement took place. If the corresponding UL RAW was also delayed for the next BI, DL RAW should not end as early as denoted by (34) because the DL packet will not be ready due to the processing time at the master node. The earliest DL RAW end in that case should occur $2t_{TX}^P + t_{proc}$ after the start time of the previous RAW assigned to the node h, and not after the previous measurement timestamp. Constraint (37) takes this into account.

$$v_{ijh}\Big(t_i^{start} + 2t_{TX}^P + t_{proc}\Big) \le d_{jh} t_j^{end} \quad \forall h \in \{0,\dots,n-1\},\ \forall i \in \{0,\dots,m-2\},$$



(a) Downlink RAW boundaries for receiving the response to packet k.



(b) In case when DL RAW for reception of response to packet k cannot fit into the same beacon interval as its corresponding UL RAW for transmission of packet k (in this case because DL RAW would have to end when the beacon interval ends, but DL packet is not ready yet because of the long processing time), it has to be scheduled in the next beacon interval. Such DL RAW can start any time t_{TX}^P before the node enqueues its next packet and end before the node enqueues its next packet.

FIGURE 5. Downlink RAW scheduling for (a) the first packet in a BI and (b) the last packet in a BI whose response cannot be delivered in that BI.

$$\forall j \in \{i+1, \ldots, m-1\} \tag{36}$$

$$d_{ih}\left(t_h^{RAW,prev} + 2t_{TX}^P + t_{proc}\right) \leq d_{ih} t_i^{end} \quad \forall h \in \{0, \ldots, n-1\},\ i \in \{0, \ldots, m-1\}. \tag{37}$$

One of the constrains in (34) or (37) will be stricter and ensure that there is enough time after packet transmission to compute the response and deliver it back to the node.

Only 7 out of 13 decision variables listed in Table 2 are mathematically necessary. The other 6 are auxiliary variables which could be omitted as they are defined by equality constrains (i.e. (11), (15), (19), (20), (21), (22)). Auxiliary variables are introduced either for improving clarity or to reduce third order constrains to quadratic.

VI. PERFORMANCE EVALUATION

An experimental evaluation of the proposed scheduling approach is performed in simulation. The IEEE 802.11ah module in the ns-3 network simulator is used to model different network scenarios [37]. The Gurobi optimizer, linked to ns-3, is used to compute the optimal solution of the INP [38]. Three performance aspects of the INP are analyzed in this section. First and foremost, we examined how well it schedules and assigns RAWs to the slave nodes by measuring how often they do (not) meet the deadline considering cycle times in range 50 ms–500 ms. Secondly, we examined the impact of such scheduling on the other nodes sharing the network with slave nodes. Finally, we analyzed the performance of the algorithm itself, namely the computational complexity and feasibility. The simulation setup is outlined in Subsection VI-A, whereas Subsection VI-B presents an overview of the evaluation results.

TABLE 3. Default parameters used in the simulations.

PHY parameters	**Values**
Frequency [MHz]	868
TX power [dBm]	0
TX/RX gain [dB]	0
Noise Figure [dB]	6.8
Coding method	BCC
Error rate model	YansErrorRate [40]
MAC parameters	**Values**
Duration of AIFS (μs)	316
Cross slot boundary	disabled
Station distribution	random
Rate control algorithm	constant
Max size of transmit queue (packets)	10
Packet size (bytes)	130
Payload size (bytes)	53
Total number of nodes in the network N	50
Reporting interval (s) of non-slaves	1
Number of slave nodes	1, 2, 3, 4
Control loop cycle-time T_h^{cyc} (ms)	$50 \leq T_h^{cyc} \leq 512$
Ethernet transmission delay (μs)	6.56
Ethernet data rate (Mbps)	100
Data mode (2 MHz)	MCS8
Topology radius (m)	100
Simulation time (s)	100
Input to the INP	**Values**
Beacon interval T_{BI} (μs)	102400
Max. channel time T_{CH}^{max} (μs)	100360
Time reserved for packet transmission t_{TX}^P (ms)	3
Processing time t_{proc} (ms)	5
Time reserved for next beacon t_{BI}^{TX} (ms)	5
Max. number of RAWs m	$4n \cdot 2 + 1$

A. SETUP

The baseline configuration shown in Table 3 is used to analyze the performance of closed loop communication with IEEE 802.11ah. We used a standard log propagation loss model with values for outdoors scenarios and macro deployment [39], [40]. Given that IEEE 802.11ah is an IP-based technology, we have adopted the open Internet Engineering Task Force (IETF) stack for embedded devices (CoAP, UDP, IPv4/6) to implement the interactions between slaves and the master. A 53-byte payload is carried in 130-byte packets, considering the overhead of all protocol headers (802.11ah, IPv4, UDP, CoAP).

Although an IEEE 802.11ah AP can support up to 8192 connected stations, we consider a small number of simultaneously operating non-slave nodes, assuming that the AP uses some standard mechanisms to decrease the contention between them in dense or ultra-dense deployments [20], [21], [24], [41]. Reporting interval of the non-slaves is set to 1 s, which faithfully represents traffic patterns of many more stations in real deployments. Non-critical sensors in practice do not report often, progress reporting sensors may report every few days, majority of sensors will report much less frequently than simulated, hence the amount of total traffic of N simulated nodes faithfully reflects the amount of traffic in practice. Similarly, we only test fast control loops and amount of traffic that a single control loop with 50 ms cycle time produces equals the amount of traffic 10 control loops with 500 ms cycle time. For the sake of comparison, remember that a commercial WirelessHART AP can support up to 8 devices reporting every 500 ms [16]. IEEE 802.11ah AP computes the optimal RAW configuration for slave nodes using INP every beacon interval. All *empty* RAWs in the output of the INP are subsequently assigned to non-slave nodes, each one transmitting one 130-byte packet per second. The input parameters that INP uses are updated every beacon interval for every slave node, except the constant parameters shown in Table 3.

We conducted experiments for various control-loop cycle-times, both shorter and longer than the set BI, in both static and dynamically changing networks using MCS8. In a star topology, nodes are randomly positioned in a circle with a radius of 100 m around the AP. Nodes are static and we did not consider mobility or variations of the channel model. The master node is wired via Ethernet to the AP. We varied the number of slave-nodes n in the network, while keeping 50 nodes in the network in total. The maximal number of allowed RAWs m depends on the number of slave nodes, more slave nodes require more slots in the solution. Each experiment was repeated 10 times with different random seeds and the results in Section VI-B show the average values of 10 randomized simulations.

The mixed integer program is implemented in C++, using version 8.1.1 of the Gurobi library. Simulations were run on a HP ProLiant XL170r Gen9 server with 20 Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40 GHz processors and 64 GB of RAM.

B. RESULTS

The network performance with INP is compared against the RBI method, as it is the only preexisting RAW reconfiguration strategy that takes into account both uplink and downlink traffic [4]. Due to the randomized scheduling, the value of input parameters and demanding traffic requirements, in some BIs INP was infeasible. This infeasibility indicates that the network cannot support the traffic demands along with the strict timing constrains. When the infeasible instances are below 2%, the mean packet deadline-miss ratio is 0.417% with 0.3375% standard deviation. Lost packets also miss the deadline, hence are included in this statistic. However, the percentage of lost packets never surpasses 0.05%. In case of infeasibility, no RAW is present and all nodes contend for the medium (default IEEE 802.11ah operation). As the above mentioned deadline-miss ratio shows, in case of infeasibility (i.e. with the default IEEE 802.11ah non-optimized MAC) some packets still get successfully sent and delivered before the deadline, but most of them do not.

As is evident from Fig. 6, RBI method can result in a slightly better deadline-miss ratio than INP in a static network, when scheduled carefully. This is due to over dimensioning the channel access opportunities with RBI, i.e. slave nodes have more allocated channel time than necessary due

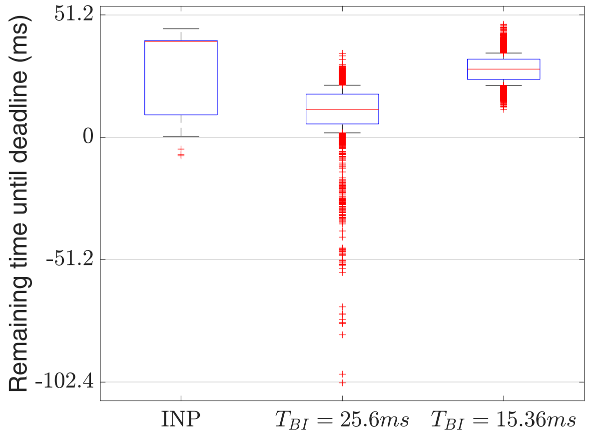
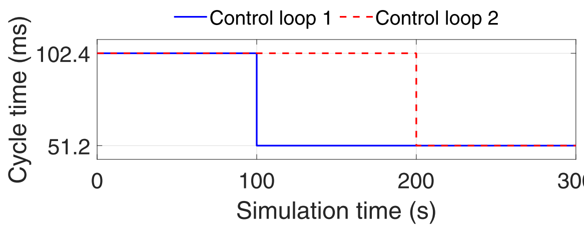


FIGURE 6. In low-density cases where RBI method can successfully over dimension channel access for slaves, it can yield better results than INP regarding the deadline-miss ratio. In a single CL with T_h^{cyc} = 52.1 ms, all packets meet the deadline with the RBI method using the shortest possible T_{BI} for this scenario, whereas 0.0512% of packets miss the deadline with INP.

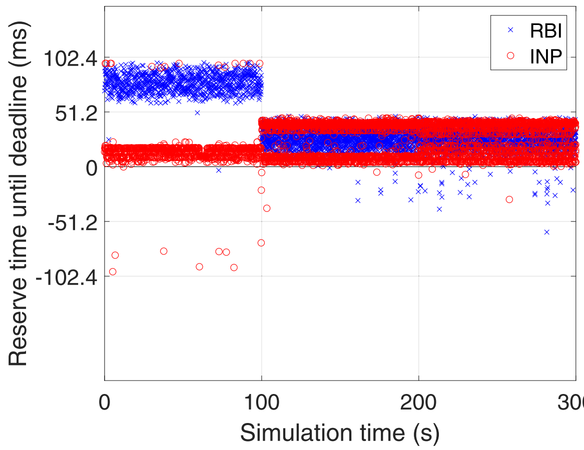
to the beacon interval reduction and non-congested medium. However, only the shortest possible T_{BI} of 15.36 ms that can theoretically accommodate the control loop traffic in the network in this scenario missed fewer deadlines than INP, whereas larger beacon intervals do not provide sufficient over dimensioning of channel access and missed more deadlines. However, beaconing every 15.36 ms needs 51.5542 kbps of bandwidth on average, versus only 13.5691 kbps with INP. For the sake of comparison, a CL exchanging 64-byte payload every $T_h^{cyc} = 51.2$ ms needs 20.3125 kbps.

Dynamically changing networks with CLs that have variable cycle times generally have worse deadline-miss ratio than INP. A simple example of a dynamic network with 48 UL-transmitting sensors and 2 CLs that change their respective T_h^{cyc} during the experiment according to the rule in Fig. 7a results in 0.88% and 7.24% deadline-miss ratios, whereas INP results in 0.34% and 0.33%, as illustrated in Fig. 7.

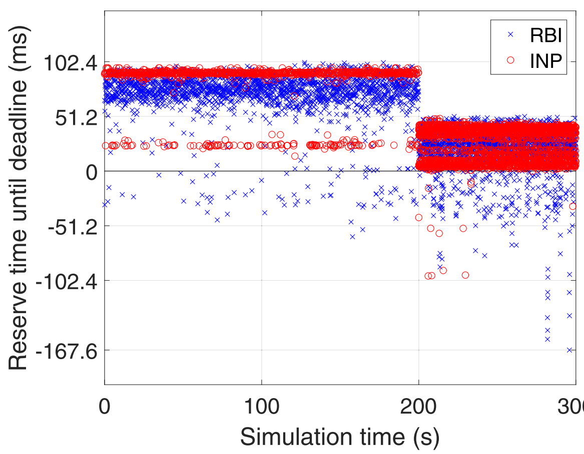
The scheduling performance of INP is compared to the one of RBI method in Fig. 8 and Fig. 9. Fig. 8a illustrates the percentage of late packets for varying number of slave nodes which implement control loops for 50 ms and 200 ms cycle. As the number of slave nodes increases, the INP performs better than the RBI method. To evaluate the performance of INP for various cycle times, we compare the deadline miss ratio of 1 and 3 slave nodes in Fig. 9. For a single control loop (cf. Fig. 9a), both methods miss very few deadlines (<0.3542%), but INP performs worse than RBI method for short cycle times due to 1%–2% infeasible cases. However, it should be noted that for short cycles of 50 ms, 100 ms and 150 ms RBI method requires shortening the beacon interval to 15.36 ms, 40.96 ms and 61.44 ms respectively, thus occupying 6.92, 2.76 and 1.83 times more channel bandwidth than INP for beaconing. INP could support two additional control loops with $T_h^{cyc} = 50$ ms for the difference in bandwidth occupied by beaconing in the RBI method. Already for three control



(a) Cycle time set points for an experiment with two CLs in a dynamically changing network.

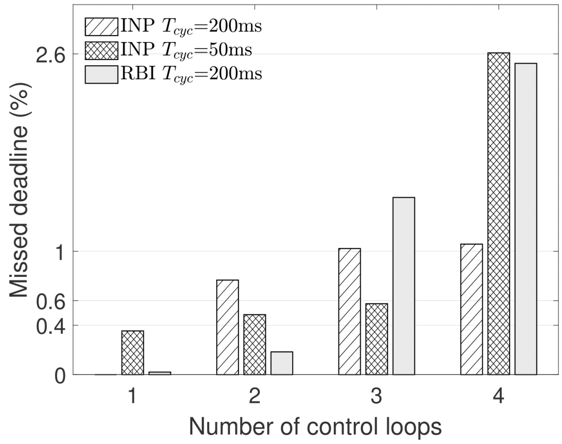


(b) CL 1: 0.34% and 0.88% packets are late with INP and RBI method respectively.

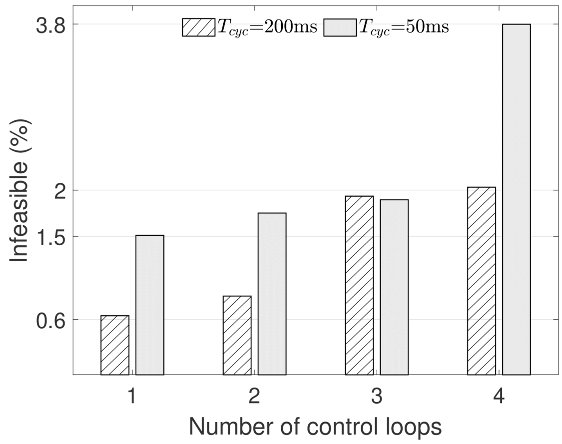


(c) CL 2: 0.33% and 7.24% packets are late with INP and RBI method respectively.

FIGURE 7. INP performs better than the best case RBI method in an example of a dynamically changing network containing two control loops and 48 non-slave nodes.



(a) Total percentage of control packets that missed the delivery deadline. RBI for $T_h^{cyc} = 50$ ms is omitted for clarity (it results in $< 0.25\%$ missed deadlines for 1 and 2 loops, but 17.9% and 39.6% for 3 and 4 loops respectively).



(b) Percentage of beacon periods where INP is infeasible in the experiment (no RAW grouping).

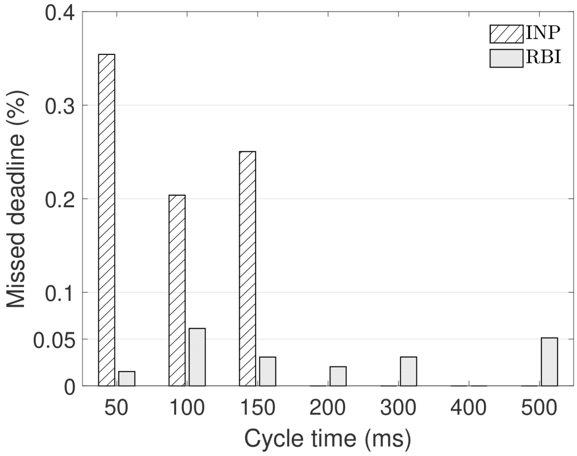
FIGURE 8. Performance of INP and RBI for various number of control loops with $T_h^{cyc} = \{50, 200\}$ ms.

loops, INP misses much fewer deadlines than the RBI method for short cycle times, as illustrated in Fig. 8a.

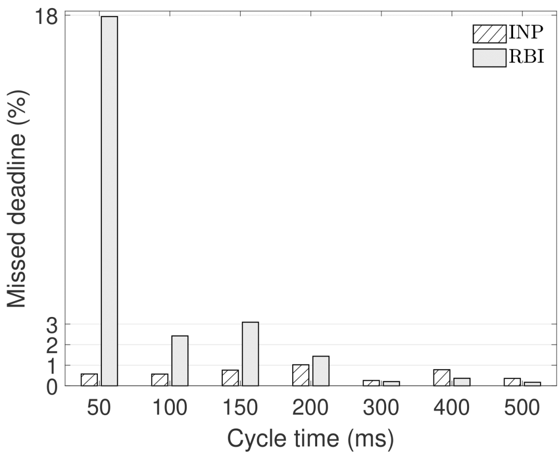
The impact of both INP and RBI methods on the non-slave nodes is illustrated in Fig. 10. Already for 3 loops with $T_{cyc} = 51.2$ ms, RBI method no longer has sufficient channel time for the 47 non-slave nodes due to the frequent beaconing, resulting in severe losses of their packets (cf. Fig. 10c). INP enables more efficient usage of channel bandwidth and does not add as much overhead to the beacons as RBI method, given the adaptive RAW configuration. Hence, it saves energy both for slave and non-slave nodes, as shown in Fig. 10b. In non-saturated conditions, when RBI method is able to over dimension channel access for slaves, non-slave nodes consume slightly less energy than with INP due to less idle time and not significantly more RX time (beacon reception). On the other hand with INP, slave nodes use less channel time and energy, leaving more time to non-slave nodes to be idle given that INP only optimizes RAW configuration for slave nodes.

VII. COMPLIANCE TO SPECTRUM REGULATIONS

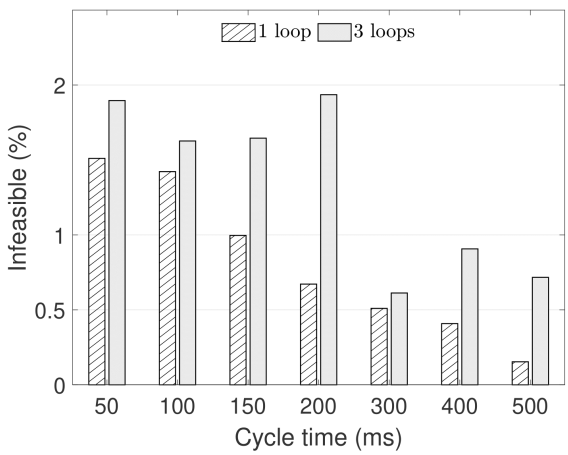
IEEE 802.11ah operates in unlicensed 863-868 MHz bands (Europe) that are subject to regulations of the fair spectrum usage. Each band imposes limits to the maximum amount of time devices are allowed to transmit. These limits



(a) 1 control loop

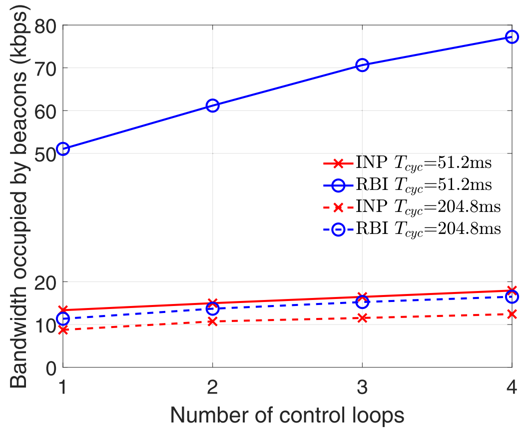


(b) 3 control loops

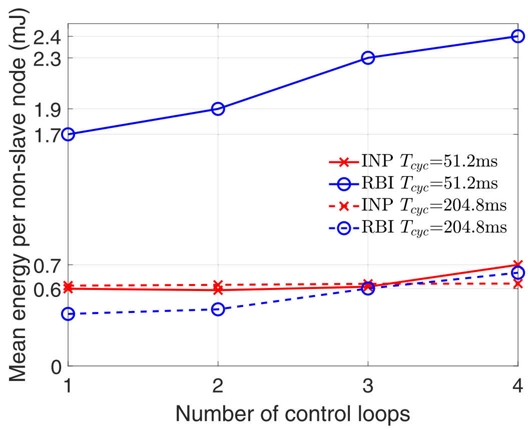


(c) Percentage of beacon periods where INP is infeasible in the experiment (no RAW grouping).

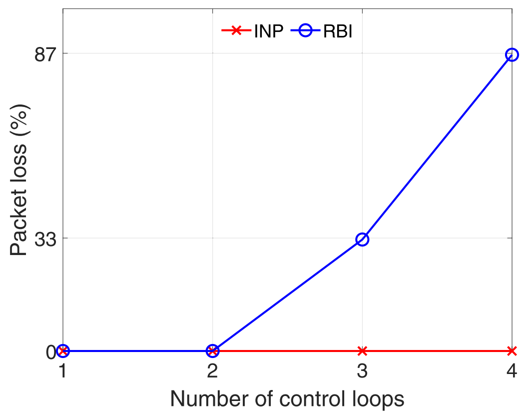
FIGURE 9. Performance of INP and RBI method for 1 and 3 control loops with various cycle times.



(a) Increasing the number of control loops increases the RPS size, hence it increases the beacon overhead, wasting the channel bandwidth for useful traffic.



(b) Increasing the number of control loops leaves less channel time for non-slave nodes, thus they sleep more. It also increases the number of retransmissions if there is not enough channel bandwidth for non-slave nodes, increasing their energy consumption nevertheless.



(c) For more than two loops with T_{cyc}=51.2 ms, there is not enough channel bandwidth left for non-slave nodes reporting every second.

FIGURE 10. Influence of INP and RBI method on non-slave nodes in the network.

are specified either in the form of a duty cycle or in the form of restrictions concerning polite spectrum access [5]. Duty cycle is the ratio of the cumulative sum of TX time per observation period fixed to one hour in Europe [6]. The regulator specified the duty cycle of 0.1% (3.6s per 1h) in 863–865 MHz bands, and 1% (36s per 1h) in 865-868 MHz bands. How the TXs are spread in time is not restricted. For example, a 1% duty cycle allows devices to transmit 36 s continuously and be quiet for the remaining 3600 s–36 s, as well as to transmit for 3.6 s 10 times per hour. Also, only TX times of the TXs within a particular frequency band are included in the limitation. Therefore, TXs can occur in multiple bands simultaneously, allowing a device to sequentially transmit in different bands and maximize its allowed TX time [5].

IEEE 802.11ah implements polite spectrum access methods compliant to the regulation, hence it enjoys somewhat loosened restrictions compared to the 1% duty cycle otherwise enforced in its bands. Polite spectrum access implies two mechanisms: Listen Before Talk (LBT) and Adaptive Frequency Agility (AFA). LBT enforces the devices to listen if the medium is busy using a Clear Channel Assessment (CCA) check for at least 160 μs. In case of a busy medium, the device must wait a random backoff interval or change the frequency before the next check (AFA). When polite spectrum access is in use, the channel use is not restricted by the aforementioned duty cycles, but with the following:

- Minimum T-OFF time of 100 ms – the minimal time that the transmitter needs to be quiet for in between its subsequent transmissions.
- Maximum continuous TX-ON time of 1 s for a single transmission, and 4 s for multiple transmissions separated by interval smaller than 5 ms, as part of a bidirectional protocol – the maximum time that a transmitter can be actively emitting continuously.
- Maximum cumulative TX-ON time of 100 s/1 h over 200 KHz of the spectrum.

Minimum T-OFF time restricts cycle times to at least 100 ms+t_{TX}, where t_{TX} is the transmission time defined by (39). However, more frequent transmissions in different channels could still be allowed, assuming that the number of slave nodes is limited to at most (100 ms+t_{TX})/(5t_{TX}). For example, at most 38 control loops with minimal compliant cycle time of $T_h^{cyc} = 20.104$ ms hopping over all five 1 MHz channels using MCS8 could be supported. Maximum Continuous TX-ON does not influence control loops given their short packets and cyclic scheduling. Maximum cumulative TX-ON time imposes the minimal cycle time T_h^{cyc} of a regulation-compliant Wi-Fi HaLow control loop as follows (cf. Table 4):

$$\frac{t_{TX}}{T_h^{cyc}} \leq \frac{100s}{3600s} \implies T_h^{cyc} \geq 36 \cdot t_{TX}, \tag{38}$$

where T_h^{cyc} is the cycle time of a slave node h.

The duration of a single Wi-Fi HaLow packet transmission t_{TX} with a long preamble is

$$t_{TX} = 320\mu s + 40\mu s \cdot \left\lceil \frac{14bits + 8 \cdot l}{DR \cdot 40\mu s} \right\rceil, \tag{39}$$

where l is the packet size in bytes and DR is the data rate.

To respect the minimum T-OFF time in each of the five available channels, a station may only emit once every 20.104 ms using MCS8. More than double data rates are achievable on 2 MHz channels, enabling faster transmissions and lower cycle times. However, only two 2 MHz channels are disposable for hopping, making 50 ms+$t_{TX}/2$ the smallest cycle time compliant with the T-OFF constraint. At most 125 control loops are allowed with the smallest cycle time on 2 MHz channel with MCS8. For 1 MHz channels, regulation compliant cycle times range from 20.104 ms for 8-byte payload via MCS8 up to 324 ms for 256-byte payload via MCS0.

TABLE 4. Minimum T-OFF time compliant cycle time in ms as a function of data rate and payload for 1 MHz channel bandwidth.

Data Rate (Kbps) \ Payload (bytes)	8	16	32	64	100	128	256
300	86.4	93.6	109.4	139.68	174.24	201.6	324
600	49	53.3	60.5	76.32	93.6	106.56	168.48
900	37.4	38.9	44.7	54.72	66.24	74.88	116.64
1200	30.2	33.1	36	44.64	53.28	59.04	90.72
1800	24.5	25.9	28.8	33.12	38.88	43.2	64.8
2400	21.6	23	24.5	28.8	33.12	36	51.84
2700	20.2	21.6	23	25.92	30.24	33.12	47.52
3000	20.2	20.2	21.6	24.48	28.8	31.68	43.2
3600	18.7	18.7	20.2	23.04	25.92	27.36	38.88

VIII. CONCLUSION

In this article we present a detailed Integer Nonlinear Problem (INP) problem formulation for precise scheduling of control loops. INP is designed to minimize the timeshare of the channel reserved for control loop end nodes, while satisfying their stringent timing requirements. We used commercial Gurobi solver to find the optimal INP solution in every beacon interval. In a dynamically changing network hosting control loops with 51.2 ms and 102.4 ms cycles, the INP solution fails to meet the deadlines in 0.335% cases on average, whereas it fails to meet it in 4.06% cases in [4], introducing a significant improvement. The improvement in comparison to our previous work [4] increases proportionally to the traffic demands in the network, whereas in the networks with very low traffic demands this algorithm performs similarly to [4]. Furthermore, this article demonstrates better scalability of IEEE 802.11ah than well-established industrial wireless technologies WirelessHART and ISA100.11a. The INP solution presents the optimal RAW configuration for slave nodes and groups them accordingly, without any apriori knowledge of the network. This model is adaptive to real-time changes in the network where nodes join and leave the network, as well as in case of variable cycle time of a single node. The design offers guarantees for critical traffic, sacrificing capacity for non-critical traffic. Hence, for the latter, there may be more contention and more energy consumption. However, numerous throughput enhancement and energy conservation mechanisms for non-critical sensors nodes are present in the literature, whereas very limited research has been conducted so far for scenarios that include actuators. We focus on networks with limited number of control loops coexisting with a large number of non-critical sensors that do not impact the loops. Finally, solving the INP problem is computationally complex. Gurobi can compute the optimal

RAW configuration in less than 10 ms only for a few control loops with large cycle times (100ms or more), which limits its real-time usage to such scenarios. Computation time would be too large for this solution to operate in real-time in presence of more than 4 control loops, any of them having less than 40 ms cycle time. However, commercial solvers can be used to find optimal RAW configurations for different scenarios offline and feed them to the AP during runtime, or run on a powerful external processor at the AP. We hope that this INP problem formulation can inspire the research for adequate less-complex algorithms that can operate in real-time in all conditions, and serve as a baseline for heuristic optimization or other approximate methods.

REFERENCES

[1] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam, "The wireless control network: A new approach for control over networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2305–2318, Oct. 2011.

[2] D. Cavalcanti, J. Perez-Ramirez, M. M. Rashid, J. Fang, M. Galeev, and K. B. Stanton, "Extending accurate time distribution and timeliness capabilities over the air to enable future wireless industrial automation systems," *Proc. IEEE*, vol. 107, no. 6, pp. 1132–1152, Jun. 2019.

[3] H. Hellstrom, M. Luvisotto, R. Jansson, and Z. Pang, "Software-defined wireless communication for industrial control: A realistic approach," *IEEE Ind. Electron. Mag.*, vol. 13, no. 4, pp. 31–37, Dec. 2019.

[4] A. Seferagic, I. Moerman, E. De Poorter, and J. Hoebeke, "Evaluating the suitability of IEEE 802.11ah for low-latency time-critical control loops," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7839–7848, Oct. 2019.

[5] M. Saelens, J. Hoebeke, A. Shahid, and E. D. Poorter, "Impact of EU duty cycle and transmission power limitations for sub-GHz LPWAN SRDs: An overview and future challenges," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 219, Dec. 2019.

[6] D. Castells-Rufas, A. Galin-Pons, and J. Carrabina, "The regulation of unlicensed sub-GHz bands: Are stronger restrictions required for LPWAN-based IoT success?" 2018, *arXiv:1812.00031*. [Online]. Available: http://arxiv.org/abs/1812.00031

[7] V. K. L. Huang, Z. Pang, C.-J.-A. Chen, and K. F. Tsang, "New trends in the practical deployment of industrial wireless: From noncritical to critical use cases," *IEEE Ind. Electron. Mag.*, vol. 12, no. 2, pp. 50–58, Jun. 2018.

[8] Z. Pang, M. Luvisotto, and D. Dzung, "Wireless high-performance communications: The challenges and opportunities of a new target," *IEEE Ind. Electron. Mag.*, vol. 11, no. 3, pp. 20–25, Sep. 2017.

[9] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolie, "Design of a low-latency, high-reliability wireless communication system for control applications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 3829–3835.

[10] Y.-H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, M. Tomizuka, T. Li, D. Malone, and D. Leith, "RT-WiFi: Real-time high speed communication protocol for wireless control systems," *ACM SIGBED Rev.*, vol. 10, no. 2, p. 28, 2013.

[11] O. Seijo, I. Val, and J. A. Lopez-Fernandez, "W-SHARP: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications," *IEEE Trans. Ind. Informat.*, early access, Jun. 7, 2020, doi: 10.1109/TII.2020.3007323.

[12] *Industrial Networks: Wireless Communication Network and Communication Profiles*, Standard IEC PAS 62948, 2015.

[13] J. Sachs, G. Wikstrom, T. Dudda, R. Baldemair, and K. Kittichokechai, "5G radio network design for ultra-reliable low-latency communication," *IEEE Netw.*, vol. 32, no. 2, pp. 24–31, Mar. 2018.

[14] *HART Field Communication Protocol Specification HFC_SPEC12, Revision*, HART Commun. Found., FieldComm Group, Austin, TX, USA, vol. 6, 2006.

[15] *Wireless Systems for Industrial Automation: Process Control and Related Applications*, Standard ISA-100.11 a-2009, International Society of Automation (ISA) Standard, 2009.

[16] (2020). *One Wireless Field Device Access Point Specification Release 310 OW03-650-310 Kernel Description*. Accessed: Nov. 9, 2020. [Online]. Available: https://www.honeywellprocess.com/library/marketing/tech-specs/onewireless-fdap-specification.pdf

[17] A. Seferagic, J. Famaey, E. D. Poorter, and J. Hoebeke, "Survey on wireless technology trade-offs for the industrial Internet of Things," *Sensors*, vol. 20, no. 2, p. 488, Jan. 2020.

[18] A. Šljivo, D. Kerkhove, L. Tian, J. Famaey, A. Munteanu, I. Moerman, J. Hoebeke, and E. D. Poorter, "Performance evaluation of IEEE 802.11ah networks with high-throughput bidirectional traffic," *Sensors*, vol. 18, no. 2, p. 325, Jan. 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/2/325

[19] A. Bel, T. Adame, and B. Bellalta, "An energy consumption model for IEEE 802.11ah WLANs," *Ad Hoc Netw.*, vol. 72, pp. 14–26, Apr. 2018.

[20] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "IEEE 802.11AH: The WiFi approach for M2M communications," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 144–152, Dec. 2014.

[21] A. Kureev, D. Bankov, E. Khorov, and A. Lyakhov, "Improving efficiency of heterogeneous Wi-Fi networks with joint usage of TIM segmentation and restricted access window," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.

[22] T. Kim and J. M. Chang, "Enhanced power saving mechanism for large-scale 802.11ah wireless sensor networks," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 4, pp. 516–527, Dec. 2017.

[23] B. Badihi, L. F. D. Carpio, P. Amin, A. Larmo, M. Lopez, and D. Denteneer, "Performance evaluation of IEEE 802.11ah actuators," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.

[24] L. Tian, J. Famaey, and S. Latre, "Evaluation of the IEEE 802.11ah restricted access window mechanism for dense IoT networks," in *Proc. IEEE 17th Int. Symp. A World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2016, pp. 1–9.

[25] C. Kai, J. Zhang, X. Zhang, and W. Huang, "Energy-efficient sensor grouping for IEEE 802.11ah networks with max-min fairness guarantees," *IEEE Access*, vol. 7, pp. 102284–102294, Jul. 2019.

[26] T. Kim, D. Qiao, and W. Choi, "Energy-efficient scheduling of Internet of Things devices for environment monitoring applications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.

[27] H. Mosavat-Jahromi, Y. Li, and L. Cai, "A throughput fairness-based grouping strategy for dense IEEE 802.11ah networks," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 1–6.

[28] L. R. Lakshmi and B. Sikdar, "Achieving fairness in IEEE 802.11ah networks for IoT applications with different requirements," in *Proc. ICC-IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[29] T.-C. Chang, C.-H. Lin, K. C.-J. Lin, and W.-T. Chen, "Load-balanced sensor grouping for IEEE 802.11ah networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 1–6.

[30] C.-C. Hu, Z.-B. Liu, Y.-M. Lin, and G.-J. Xu, "An efficient and effective regrouping algorithm for minimizing hidden pairs in 802.11ah networks," in *Proc. Int. Conf. Adv. Comput. Technol., Inf. Sci. Commun.*, 2019, pp. 191–196.

[31] C.-C. Hu, "Approximation algorithms of minimizing hidden pairs in 802.11ah networks," *IEEE Access*, vol. 7, pp. 170742–170752, Nov. 2019.

[32] E. Khorov, A. Lyakhov, I. Nasedkin, R. Yusupov, I. F. Akyildiz, and J. Famaey, "Fast and reliable alert delivery in mission-critical Wi-Fi HaLow sensor networks," *IEEE Access*, vol. 8, pp. 14302–14313, Jan. 2020.

[33] NEWRACOM Inc. (2019). *NEWRACOM Wins the Best Wi-Fi IoT Product 2019 Award at Wi-Fi now*. [Online]. Available: https://www.prnewswire.com/news-releases/newracom-wins-the-best-wi-fi-iot-product-2019-award-at-wi-fi-now-300957015.html

[34] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation*, IEEE Standard 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016), May 2017, pp. 1–594.

[35] W. Liang, M. Zheng, J. Zhang, H. Shi, H. Yu, Y. Yang, S. Liu, W. Yang, and X. Zhao, "WIA-FA and its applications to digital factory: A wireless network solution for factory automation," *Proc. IEEE*, vol. 107, no. 6, pp. 1053–1073, Jun. 2019.

[36] S. Tschimben, K. Gifford, and R. Brown, "IEEE 802.11ah SDR implementation and range evaluation," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[37] L. Tian, A. Šljivo, S. Santi, E. De Poorter, J. Hoebeke, and J. Famaey, "Extension of the IEEE 802.11ah ns-3 simulation module," in *Proc. 10th Workshop Ns-3*, 2018, pp. 53–60.

[38] *Gurobi Optimizer Reference Manual*. Accessed: Dec. 12, 2019. [Online]. Available: https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/9.0/refman.pdf

[39] A. Hazmi, J. Rinne, and M. Valkama, ''Feasibility study of IEEE 802.11 ah radio technology for IoT and M2M use cases,'' in *Proc. IEEE Globecom Workshops*, Dec. 2012, pp. 1687–1692.

[40] L. Tian, S. Deronne, S. Latré, and J. Famaey, ''Implementation and validation of an IEEE 802.11ah module for ns-3,'' in *Proc. Workshop ns-3*, 2016, pp. 49–56.

[41] S. Bhandari, S. K. Sharma, and X. Wang, ''Device grouping for fast and efficient channel access in IEEE 802.11ah based IoT networks,'' in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

AMINA SEFERAGIĆ received the M.S. degree in electrical engineering (automatic control and electronics) from the University of Sarajevo, Bosnia and Herzegovina, in 2016. She is currently pursuing the Ph.D. degree in computer science engineering with Ghent University, Belgium. In 2017, she was an Intern in robotics with the Corporate Research Center of Robert Bosch GmBH, Renningen, Germany. She is currently with the Internet Technology and Data Science Laboratory, IMEC, a core research group, research and development and innovation hub in nanoelectronics and digital technologies with research activities embedded in Ghent University, Belgium, and the University of Antwerp, Belgium. She has authored or co-authored several publications in international journals or conference proceedings. Her research focuses on protocol and algorithm design for wireless communications, coexistence and optimization of wireless networks with focus on sub-GHz technologies, and Internet of Things applications. She has received three awards during her career, of which two Golden Badge awards from the University of Sarajevo and a Best Paper Award from the WNS3 2018.

ELI DE POORTER received the M.S. degree in computer science engineering from Ghent University, Belgium, in 2006, and the Ph.D. degree from the Department of Information Technology, Ghent University, in 2011. Since 2015, he has been a Professor with the IDLab Research Group, Ghent University, and IMEC, where he is currently involved in and/or research coordinator of multiple national and international projects. He is currently a Professor with Ghent University. He has authored or co-authored more than 150 articles published in international journals or in the proceedings of international conferences. His main research interests include wireless network protocols, IoT networks, network architectures, wireless sensor and ad hoc networks, future Internet, self-learning networks, and next-generation network architectures. He is part of the program committee of several conferences.

JEROEN HOEBEKE received the M.S. degree in computer science engineering and the Ph.D. degree in computer science engineering with research focus on adaptive ad hoc routing and virtual private ad hoc networks from Ghent University, Belgium, in 2002 and 2007, respectively. He is currently an Associate Professor with the Internet and Data Laboratory, Department of Information Technology, Ghent University, and IMEC, Flanders. He is performing and coordinating research on wireless IoT connectivity, such as LPWAN, 802.15.4, BLE, and Wi-Fi Halow, embedded communication stacks, such as IPv6/6LoWPAN/SCHC, and deterministic wireless communication and network management, such as coexistence, determinism, scalability, and application-network interactions. This expertise has been applied in a variety of application domains, such as logistics, Industry 4.0, building automation, healthcare, and animal monitoring. He is particularly active in national funded projects and in defining, executing, and managing such projects. He has also been involved in several EU research funded projects. He has authored or co-authored more than 150 publications in international journals or conference proceedings.