Slot Bonding for Adaptive Modulations in IEEE 802.15.4e TSCH Networks

Glenn Daneels[1], Carmen Delgado[1], Robbe Elsas[2],
Eli De Poorter[2], Steven Latré[1], Chris Blondia[1], and Jeroen Famaey[1]

[1]*IDLab, University of Antwerp - imec*, Antwerp, Belgium, firstname.lastname@uantwerpen.be
[2]*IDLab, Ghent University - imec*, Ghent, Belgium, firstname.lastname@ugent.be

***Abstract*—The numerous applications of industrial automation have always posed many challenges for wireless connectivity. In the last decade, IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) networks have provided high reliability and low-power operation in such challenging industrial environments. Typically, TSCH networks employ one modulation at the physical layer and are thus limited by the characteristics of the chosen modulation in terms of, among others, data rate, reliability and energy efficiency. To tackle these limitations and to improve network performance and flexibility in those challenging industrial environments, this work explores the simultaneous use of multiple modulations in a TSCH network. Traditionally, TSCH relies on fixed-duration slots, large enough to send a packet of any size given the fixed data rate. In order to avoid wasting airtime when simultaneously using modulations with different data rates, we propose the concept of slot bonding. This allows the creation of different-sized bonded slots with a duration adapted to the data rate of each chosen modulation. To analyse the proposed slot bonding technique, we formally describe the TSCH slot bonding problem in terms of optimizing the packet delivery ratio while minimizing radio on time, with the inclusion of parent selection and interference avoidance. Afterwards, we propose a genetic algorithm that allows us to implement the problem and find solutions heuristically. Finally, we provide insights into preferred parent selection and modulation configurations by using this heuristic approach during extensive simulation experimentation in which the scalability advantage of slot bonding over longer fixed-duration slots is also shown.**

***Index Terms*—Slot bonding, TSCH, wireless mesh networks, IEEE 802.15.4e**

I. Introduction

Modern industry and its numerous applications, ranging from time-critical remote control of actuators to continuous monitoring of machinery, pose hard challenges for wireless connectivity [1]. The stringent industrial requirements are often defined by a trade-off between low delay, high reliability and low-power operation. To tackle the challenges of such applications, the IEEE 802.15.4 standard [2], first released in 2003, defines the physical (PHY) and Medium Access Control (MAC) layers for low-power wireless networks in the sub-GHz and 2.4 GHz frequency band. At the MAC layer, the IEEE 802.15.4e-2012 amendment, proposed new features which are very suitable for industrial applications such as the Time-Slotted Channel Hopping (TSCH) mode to combat external interference and multi-path propagation effects in challenging wireless environments [3]. In a tightly-synchronized time-slotted schedule, it uses frequency diversity to provide high reliability while maintaining low-power operation. TSCH has been the basis for the IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) standardization group that aims to adopt IPv6 in industrial standards [4]. The IEEE 802.15.4g-2012 PHY amendment introduced, among others, a set of so-called Smart Utility Network (SUN) modulations, being Frequency Shift Keying (FSK), Offset-Quadrature Phase Shift Keying (O-QPSK) and Orthogonal Frequency Division Multiplexing (OFDM), each with its own characteristics in terms of communication range, bandwidth, data rate, energy consumption and reliability [5].

For many years, TSCH has proven its efficiency in various low-power wireless mesh scenarios. However, traditionally all nodes in the network use the same IEEE 802.15.4 PHY modulation and is thus limited by the characteristics of the chosen modulation. While often being deployed in challenging wireless environments such as large industrial sites with blocking metal constructions disrupting the wireless connection, it is shown that enabling adaptive modulation switching can improve reliability in industrial environments with location-dependent heterogeneous propagation behaviour [6]. Therefore, a TSCH network could significantly improve from being able to use different modulations for different network links simultaneously and adapt the modulation of each link to the local propagation characteristics and the application's requirements.

In this work we continue with the slot bonding technique introduced in our previous work [7] that allows the use of multiple modulation and coding schemes (MCSs) in a single TSCH network. The proposed technique allows the creation of different-sized bonded slots with a duration adapted to the data rate of each chosen modulation. Slot bonding thus facilitates the use of multiple modulations within a single TSCH network in a resource-efficient manner. In contrast, traditional TSCH relies on fixed-duration slots, large enough to send a packet of any size given the fixed data rate, which results in wasted airtime if different modulations with different data rates are used simultaneously.

The contributions of this work compared to our previous work are three-fold [7]. First, we give an updated formal description of the TSCH slot bonding problem that allows us to analyse the proposed slot bonding technique, extending our previous work with interference avoidance and parent selection to optimize the overall network packet delivery ratio (PDR) while keeping the radio on time minimized. This problem has a high computational complexity when solving it for large network scenarios. Additionally, to allow a node to allocate an arbitrary number of slots to its parent, resulting in more flexible and realistic TSCH schedule allocations (compared to our previous work), a Markov chain model was introduced. To be able to implement and solve the formulated problem with the Markov Chain, we used a genetic algorithm (GA) to find the best heuristic solution for the slot bonding problem by selecting the most appropriate parent, modulation, and bonded time slots for each node. Also, because of this heuristic approach, we are able to scale up the network topology. Finally, we provide insights into preferred parent selection and modulation configurations by using this heuristic approach during extensive TSCH simulation experiments, in which the scalability advantage of slot bonding over longer fixed-duration slots in terms of network-wide PDR is also shown.

The remainder of this article is structured as follows. First, we introduce the related work on using multiple modulations simultaneously, TSCH and other techniques to combat external interference and multi-path fading effects in Section II. Second, in Section III, we explain the concept of slot bonding. In Section IV, we formally define the slot bonding problem, while in Section V, we introduce our genetic algorithm approach to solve the slot bonding problem heuristically. Afterwards, the slot bonding approach is evaluated in Section VI. Finally, Section VII presents the conclusions of this work.

II. Related Work

Due to the popularity gain of low-power wireless networks and the first capable IEEE 802.15.4g transceivers appearing on the market, there is an increased research effort on the available modulations and their usage in different real-world scenarios. There have been experimental evaluations in several outdoor scenarios [8], [9], testing of their suitability for smart building applications [10] and the interference robustness of the different modulations, i.e., more specifically the impact on the resulting PDR and depending on the length of the transmitted packet [11]. As low-power wireless networks are being used more and more in industrial settings, there is also an increased interest, specifically in the reliability performance of SUN modulations, in such environments. Tuset-Peiró *et al.* presented a large real-world data set consisting of data of 11 nodes being deployed in an industrial warehouse transmitting packets with 3 SUN modulations [6]. The results show large variability and poor PDRs, due to multi-path propagation and external interference effects, making them not suitable for stringent industrial requirements. As such, the authors and other related studies propose packet replication and the use of multiple IEEE 802.15.4 modulations for different packets, i.e., modulation diversity, to increase network reliability [12], [13]. Similarly, J. Muñoz *et al.* hint to the use of a different OFDM MCS on a per-link basis [10].

Besides progress at the PHY layer in the IEEE 802.15.4 standard, a lot of research effort also went into increasing network reliability by the introduction of the IEEE 802.15.4e TSCH mode in 2015. T. Watteyne *et al.* showed the importance of the channel hopping feature present in TSCH, to mitigate multi-path fading effects and increase reliability of low-power wireless networks [14]. Since its introduction, numerous centralized, distributed and autonomous TSCH scheduling solutions have been proposed [15], [16], [17]. While these scheduling functions aim at finding the optimal trade-off between throughput, latency and energy consumption for the given application, their performance still heavily depends on environmental factors, such as external interference and multi-path fading effects. To cope with those effects, other techniques are proposed to improve reliability, such as channel blacklisting [18] and more recently the PAREO methodology [19] on top of TSCH, which is an application of the Internet Engineering Task Force (IETF) Reliable and Available Wireless (RAW) layer-3 approach [20] of Automatic Repeat reQuest (ARQ), Replication and Elimination (RE) and Overhearing (OH). Other research to similar time-slotted multi-channel schemes for industrial Internet-of-Things (IoT) environments proposes multi-agent deep reinforcement learning to improve the channel utilization and channel success rate [21].

While all of those techniques offer valid solutions to deal with various link conditions, they are still limited by the employed modulation. That is why, in our previous work [7] we introduced and theoretically analysed the concept of TSCH slot bonding to allow modulation diversity in a resource-efficient manner to improve the reliability in one single network. M. Brachmann *et al.* proposed a similar approach of multiple PHYs by presenting, among others, two multi-PHY designs: (i) a design where slower PHYs are scheduled to have logical slots spanning multiple real slots, and (ii) a design where the slot size is based on the slowest PHY [22]. Their multi-template design is similar to the slot bonding approach we introduced in our previous work. In this article, to further evaluate the slot bonding technique, we improve on the model by making it more realistic, adding interference avoidance and letting it also build an optimal multi-modulation network topology tree.

III. TSCH Slot Bonding

In this section, we first introduce the basic concepts of a TSCH network and parent selection. Afterwards, we explain the concept of slot bonding and introduce the different modulations that we consider in this work.

A. TSCH Schedule Matrix

In a TSCH network, all nodes are time-synchronized. Each node maintains a matrix-formatted schedule in which time is split up into fixed-duration time slots. A time slot should be long enough to send a packet of at most 127 bytes and receive an acknowledgement (ACK). The optimal duration thus depends on the data rate of the chosen PHY. Time slots are grouped in slot frames which are repeated over time, and whose size is application-dependent. The Absolute Sequence Number (ASN) defines the absolute time in TSCH, which starts with 0 at the first ever time slot and increments with 1 in every time slot that follows. A time slot in a slot frame is denoted by its time slot offset which, together with a channel offset, defines one element in the schedule, a *schedule cell*. To do channel hopping, the channel offset is used together with the ASN to pseudo-randomly select the frequency channel at which the packet in that cell will be transmitted.

The TSCH schedule instructs a node what to do at each slot in a slot frame: sleep, transmit or listen. A scheduled cell between two nodes is *dedicated* by default, i.e., it is reserved only for those two nodes. A cell can also be marked as *shared*, which means that multiple nodes can send at the same time and a back-off timer is applied to avoid collisions. A scheduled cell can be marked as a *transmit* cell, in which a node transmits a queued packet, or as a *receive* cell in which the node listens for a possible incoming packet.

B. Parent Selection

A TSCH network is usually organised in a tree topology, enabled by a routing protocol such as IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), as both are an essential part of the 6TiSCH architecture [4]. In such a tree topology, a node can have one or more children, and one or more parents. A node that has no children, is considered a leaf node, and the node in the tree that does not have any parents, is the root node of tree. In RPL, a network topology organised as a tree is called a Destination Oriented Directed Acyclic Graph (DODAG). An RPL node uses an Objective Function (OF) to calculate a Rank value for itself in the DODAG and advertises this value to other nodes in the network. This Rank is a scalar representation of the relative position of the node in the DODAG with regards to its neighbors. The calculation of this Rank depends on one or more chosen routing metrics and can also consider constraints, e.g., using hop count as a metric or constraining the latency to be below a certain threshold. Subsequently, that Rank value is used by the nodes to select their routing parent(s).

C. Slot Bonding

TSCH slot bonding allows a single TSCH network to exploit the characteristics of different available modulations (or MCSs) in the IEEE 802.15.4 standard. By choosing the most appropriate modulation, the performance of each link can be tailored to the local environmental conditions and application's requirements to maximize the overall network performance. When using different modulations in the same TSCH network, both the time and channel management of a TSCH schedule are affected and therefore we introduced both the concepts of *slot* and *channel bonding* in our previous work [7]. In this work however, we will only focus on slot bonding, by only considering modulations that require the same bandwidth. In case modulations with different bandwidth requirements would be considered, this work can be extended to also include channel bonding.

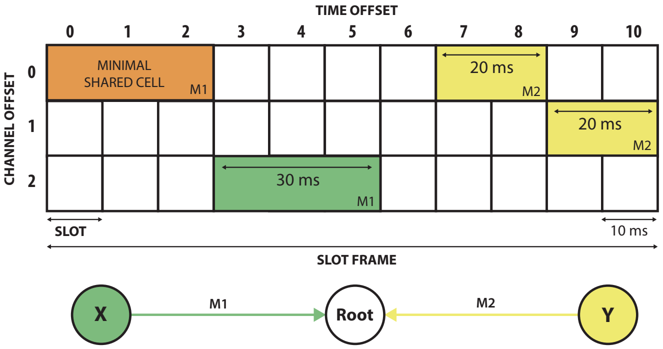


Fig. 1. A TSCH schedule with regular slots of 10 ms length. For accommodating modulation M1 on the link from node X to the root three slots are bonded to a 30 ms slot, while modulation M2 used by node Y only requires two 10 ms slots bonded together.

TABLE I
MCS2, MCS3 AND MCS4 OF OFDM OPTION 4 WHICH ALL REQUIRE 200 kHz BANDWIDTH.

Mode	Modulation, Code rate, Freq. rep.	Datarate (kbps)	Radio on (ms)	# Regular slots per bonded slot	
				10 ms	40 ms
MCS2	O-QPSK, 1/2, 2x	50	27.84	4	1
MCS3	O-QPSK, 1/2, 0	100	15.48	3	1
MCS4	O-QPSK, 3/4, 0	150	11.28	2	1

As different modulations are computationally different and have different data rates, their processing and transmission time directly impacts the optimal TSCH slot length, which should encompass the time it takes to send a data packet of the maximum allowed size (e.g., 127 bytes in TSCH) and receive an ACK. To deal with this, slot bonding bonds multiple *regular* time slots into a bonded slot which has the necessary length to transmit or receive data given the data rate and computational time of the selected modulation. This contrasts a more greedy approach in which the TSCH schedule is configured with cells that are long enough in time to be compatible with all supported modulations at once, while the proposed technique tailors each bonded slot to the requirements of the specific modulation. It thus has the advantage of limiting the waste of airtime resources.

Figure 1 shows an example of a TSCH schedule that applies slot bonding in a slot frame of 11 slots of 10 ms each. The modulation *M1* applied for the link from node X to the root requires a 30 ms bonded time slot, thus bonding 3 regular time slots. Modulation *M2* used for the link from node Y to the root requires the bonding of two 10ms time slots to a 20 ms time slot in order to be able to transmit a packet and receive an ACK. The minimal shared cell to bootstrap the network, indicated by the orange bonded slot in the figure, is configured with the same M1 modulation. Usually, the bootstrap cell should use the most robust MCS, to ensure it can be used by all nodes independent of the link quality.

While supporting multiple modulations, we make use of the flexibility of OFDM as it provides various data rates for a single bandwidth. The OFDM modulation that was introduced in the IEEE 802.15.4g-2012 amendment offers 31 PHY variations, which are subdivided in 4 different *options* that determine how many sub-carriers are grouped together in order to form an OFDM channel [11]. Each option has its own specific bandwidth requirement and also a set of different MCS values that determine how each sub-carrier is modulated (and whether frequency repetition and Forward Error Correction (FEC) are applied). As explained earlier, the different characteristics of the MCSs lead to different time slot lengths which translates to bonded slots with different numbers of regular slots for each MCS, as shown in Table I for regular slots of lengths $10\,\mathrm{ms}$ and $40\,\mathrm{ms}$. To calculate the necessary number of regular slots per bonded slot, we take into account the radio on time for a specific MCS and assume $5\,\mathrm{ms}$ of CPU processing time and $3\,\mathrm{ms}$ for reconfiguration of the radio per bonded slot, in line with the values reported by respectively Daneels *et al.* and Brachmann *et al.* [23], [22]. The radio on time for each MCS was calculated for a 127 bytes data packet and a 27 bytes ACK (i.e., the default size in TSCH to determine the slot duration), including the duration for sending the OFDM Synchronization Header (SHR) and PHY Header (PHR) in the lowest MCS of the chosen OFDM option, i.e., MCS2 of option 4 in this case [5].

Allocating a bonded slot would be very similar to the traditional 6top Protocol (6P) transaction to allocate a TSCH cell [24]. In addition to the conventional 2-step or 3-step 6P negotiation between two nodes, the nodes also agree on the employed modulation which can be specified in the reserved bits in the *CellOptions* bitmap when adding the bonded slot. When a node wants to change the modulation of a particular bonded slot, it can issue a 6P *RELOCATE* request to its neighbor that does not relocate the cell but mentions another modulation index in the *CellOptions* bitmap.

Additionally, the coexistence of slot bonding nodes and normal TSCH nodes (i.e., nodes that do not support slot bonding) in one network is possible when 2 requirements are fulfilled: (a) the regular TSCH timings are the same (i.e., the regular time slot length and all its sub-states, such as, the time to wait until transmission of the data and reception of the ACK) (b) the default modulation to join the network should also be used by the slot bonding nodes to send Enhanced Beacon (EB) messages and let other nodes join.

IV. Problem Formulation

This section formally describes the IEEE 802.15.4e TSCH slot bonding problem in which the expected number of delivered packets at the root is maximized, while the total radio on time is minimized. The inputs to this problem are the physical location of the nodes, the possible MCSs every node can employ to communicate with other nodes and the reliability of the link when using a specific MCS. As is often the case within wireless sensor network applications (e.g., periodical sensor monitoring at industrial environments), it is assumed that data packets are generated at a fixed rate. We consider interference avoidance, parent selection, MCS selection, and slot assignment.

First, we describe the expected delivered packets calculation and the radio on time calculation. Afterwards, we apply these calculations and propose the new slot bonding problem formulation. The symbols used in this section, are listed in Table II.

A. Delivered Packets Calculation

In this section we describe how for all resource allocations made in a single TSCH slot frame, the expected total number of delivered packets at the root, originating from all nodes in the network, is calculated. In our previous work, the model introduced the limitation that a node must allocate a fixed number of bonded slots for each packet present at the node. That number represented the number of times each packet could be retransmitted at most and was dependent on the link reliability from that specific node to its parent. In contrast, in this work we allow an arbitrary number of slots to be allocated to the parent, which allows for more flexible and realistic scheduling of TSCH bonded slots. In order to do so, we introduce a Markov chain model that helps us to calculate the number of packets a node can successfully transmit to its parent in a slot frame on average. Afterwards, we show how the result of the proposed stochastic process is used to calculate the expected number of delivered packets to the root of the entire network.

1) Markov Chain Model: In order to calculate the number of packets a node can successfully transmit to its parent, we propose a Markov chain model of a transmitting node in a TSCH network. This model takes as inputs the TSCH schedule slot frame with its set of slots $T = \{t_0, t_1, ..., t_{max}\}$, the number of allocated transmissions slots in the slot frame, the maximum queue size Q of the node, the number of packets in the queue at the start of the slot frame, the chosen MCS between the node and its parent that determines the reliability $l \in [0, 1]$ of the link to its parent, the maximum number of retransmissions r_{max} before the node discards the packet. Each node in the network generates a fixed number of g packets per slot frame which are assumed to be generated at the beginning of the slot frame. As for the outputs, the probabilities determined by the model allow us to calculate the probability that the node successfully sent x packets to its parent during a slot frame. More specifically, x denotes all packets that were present in the queue at the beginning of the slot frame and that are sent successfully before the end of the frame (hence leaving an empty queue). Or, x denotes that part of the packets present at the start of the frame in the queue that are sent successfully to the parent when the end of the frame is reached (i.e., leaving a non-empty queue).

Consider the stochastic process defined by the four variables (q, a, r, x), which are considered at the start of each slot t_i of a frame, $t_i \in T$, and at the end of slot t_{max}, denoted by t_{max+1}. q is the number of packets waiting for transmission with $0 \leq q \leq min(Q, q_e + g)$, with q_e being the packets left in the queue at the end of the previous slot frame. a is the number of remaining slots in the slot frame and the node can transmit

TABLE II
ALL USED SYMBOLS AND THEIR RESPECTIVE MEANING.

Symbol	Meaning
B	Matrix containing b_{v_i,v_j} probabilities at the (i, j)-th entry that represent the chain starting in transient state v_i and being absorbed in state v_j
B^l	B matrix containing $b^l_{v_i,v_j}$ probabilities calculated for a link reliability $l \in [0,1]$
C_n	Set of all children of node n, i.e., $C_n \subset N$
D_n	Set of all descendants of node n, i.e., $D_n \subset N$
F	Set of available channel offsets, i.e., $\{f_0, f_1, ..., f_{max}\}$
I_n	Set of nodes that interfere with receiving node n
J_x	Set of absorbing states of the Markov chain in which x packets arrive at the parent, i.e., $J_x \subset V$
M	Set of available MCSs, i.e., $\{m_0, m_1, ..., m_{max}\}$
N	Set of all network nodes, i.e., $\{n_0, n_1, ..., n_{max}\}$
N_0	N without the root, i.e., $\{n_1, ..., n_{max}\}$
P	Markov process transition probability matrix
Q	Max. queue size of the node
I	Identity matrix
R	Probability matrix going from transient to absorbing state
S	Set of possible number of consecutive regular slots bonded together, i.e., $S = \{1, ..., \|T\|\}$
T	Set of slots in slot frame, i.e., $\{t_0, t_1, ..., t_{max}\}$
V	Set of all Markov chain states
W	Probability matrix going from transient to transient state
a	Number of remaining slots in the slot frame which can all be used by the node for transmission
a_n	Number of transmission slots allocated by node n
b_{v_i,v_j}	The probability that the chain will be absorbed in the absorbing state v_j if it starts in the transient state v_i, is the (i,j)-th entry of the matrix B
e_n	Approx. of expected number of packets in queue of node n
g	Number of packets generated per slot frame per node
$h_{C_n,q}$	Probability of q packets arriving at n from its children C_n
l	Reliability $\in [0,1]$ of a link between node and parent, dependent on used modulation
l_n	Link reliability l for a node n to its parent
n_0	Root node of the network, i.e., $n_0 \in N$
o_{state}	Radio on time for a specific TSCH state
p_n	Parent of node n, i.e., $p_n \in N$
q	Number of packets waiting for transmission
q_e	Number of packets left in queue after the previous slot frame
r	Remaining number of retransmissions for current packet
r_{max}	Max. retransmissions before being discarded
s_m	Number of consecutive regular slots necessary for 1 bonded slot for modulation m, i.e., $s_m \in S$
u_n	Average number of transmission slots that node n is expected to use during one slot frame
v_i	State of the Markov Chain, i.e., $v_i \in V$
x	Number of packets successfully arrived at parent
$y_{x,q,a,l}$	Probability of x packets arriving at the parent at end of slot frame, while at the start of the frame the state was $(q, a, r_{max}, 0)$, over a link with reliability l
z^n_x	Probability that x packets arrive from a node to its parent
$\sigma_{t,f,s,n}$	Binary decision variable that equals 1 when node $n \in N_0$ transmits in a bonded slot, using $s \in S$ consecutive slots allocated at time offsets $\{t, t+1, \ldots, t+(s-1)\} \subset T$ and channel offset $f \in F$ to its parent $p \in N$, else 0
$\gamma_{p,m,n}$	Binary decision variable that equals 1 when node $n \in N_0$ selects $p \in N$ as its parent and $m \in M$ as the modulation to transmit to the parent, else 0

in all of them, with $a = |T| - i$ at slot t_i, $0 \leq a \leq |T|$. r is the number of remaining transmissions that are allowed for the current packet to be transmitted, $1 \leq r \leq r_{max}$. When a packet is transmitted for the first time, then $r = r_{max}$. If after a transmission that started with $r = 1$ the transmission was unsuccessful, the packet is lost. x is the number of successful packet transmissions in the previous slots of the current frame, $0 \leq x \leq |T| - a$.

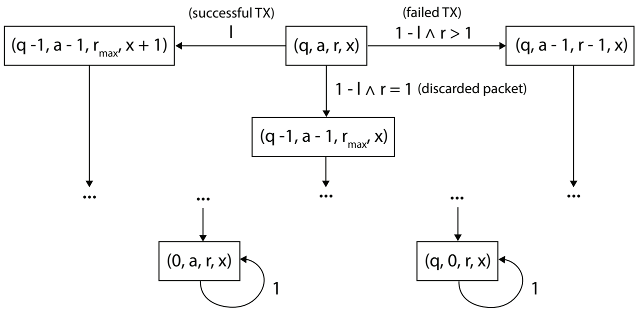


Fig. 2. Diagram of the Markov chain.

$PS(q, a, r, x)$ denotes the stochastic process and V is the set of all Markov Chain states. There are two classes of absorbing states, i.e., $(0, a, r, x)$ and $(q, 0, r, x)$, representing respectively that the process can not transition to any other state when there are no more packets left to transmit or when there are no allocated slots in the slot frame left to transmit in. In what follows we describe the possible transitions in the transition matrix P of the process PS from slot t_i, so starting from state (q, a, r, x) to slot t_{i+1}, for $0 \leq i \leq |T| - 1$, as shown in Figure 2:

- **if r > 1, a > 1 and q > 1**: the current packet is transmitted. With probability l the packet reaches the parent and the process goes to state $(q-1, a-1, r_{max}, x+1)$. With probability $1 - l$ the packet is lost and the process retries transmitting the packet in state $(q, a-1, r-1, x)$.
- **if r = 1, a > 1 and q > 1**: the current packet is retransmitted one more time. With probability l the packet reaches the parent, and process go to state $(q-1, a-1, r_{max}, x+1)$. With probability $1 - l$ the transmission fails and the packet is discarded. The process proceeds to the next packet, in state $(q-1, a-1, r_{max}, x)$.
- **if a = 0**: there are no allocated slots left for the node to transmit in, the process stays in the same absorbing state with probability 1.
- **if q = 0**: there are no waiting packets left, the process stays in the same absorbing state with probability 1.

We renumber the states of the transition matrix P in such a way that the transient states come first and the absorbing states last, resulting in following canonical form:

$$P = \begin{pmatrix} W & R \\ 0 & I \end{pmatrix} \tag{1}$$

where I is the identity matrix and W and R are non-zero matrices containing the probabilities of going from a transient state to another transient state and the probabilities going from a transient to an absorbing state, respectively.

According to the Theorem 11.6 of [25], the probability b_{v_i,v_j} that the chain will be absorbed in the absorbing state v_j if it starts in the transient state v_i, is the (i,j)-th entry of the matrix B, given $B = (I - W)^{-1} \cdot R$. Note that according to Theorem 11.4 of [25], the inverse matrix $(I - W)^{-1}$ exists.

These probabilities b_{v_i,v_j} allow us to evaluate the probability that, when the system starts in a transient state, the system is absorbed in a state $(0, a, r, x)$, i.e., the queue is emptied before the end of the frame and there have been x packets transmitted, or the system is absorbed in a state $(q, 0, r, x)$, i.e. the end of the frame is reached and there have been x packets transmitted. These probabilities are used in Section IV-A2, to calculate the average number of delivered packets at the root, for all nodes in the network, in one slot frame.

2) Number Of Delivered Packets: We describe how the expected number of packets arriving at the root in a single slot frame is calculated, using the probabilities determined by the Markov chain defined in Section IV-A1.

We define $y_{x,q,a,l}$ as the probability that at the end of the slot frame x packets have arrived at the parent when at the start of the frame the state was $(q, a, r_{max}, 0)$, over a link to the parent with reliability $l \in [0, 1]$:

$$y_{x,q,a,l} = \sum_{v_i \in J_x} b^l_{v_{start},v_i} \tag{2}$$

where $v_{start} = (q, a, r_{max}, 0)$ is the start transient state and J_x the set of absorbing states where x packets successfully arrived at the parent. The probability $b^l_{v_i,v_j} \in [0, 1]$ which is the (i, j)-th entry of the B^l matrix is calculated with the Markov chain model defined in Section IV-A1, for a link reliability $l \in [0, 1]$ from a node to its parent.

Subsequently, the probability z^n_x that x packets successfully arrive from node n to its parent is represented by:

$$z^n_x = \sum_{q=0}^{|D_n| \cdot g} h_{C_n,q} \cdot y_{x,min(Q,q+g),a_n,l_n} \tag{3}$$

where D_n is the set of all descendants of node n, a_n the number of slots allocated for transmission by n, l_n the reliability of the link to the parent of n, of which the value is dependent on propagation characteristics of the link between the node to its parent and modulation used by that node, and $h_{C_n,q}$, is the probability that in total q packets arrived from the children C_n at node n. In the case that node n is a leaf node, $h_{C_n,q}$ is considered 1. Note that $x \leq |D_n| \cdot g + g$, as in the other cases z^n_x equals 0 because a node can not transmit more packets than it received from its children, i.e., $|D_n| \cdot g$, and the packets it generates itself, i.e., g packets. $min(Q, q+g)$ is the maximum number of packets that can arrive at the parent of n from node n, with the maximum queue size being Q and $q + g$ being the number of packets q from the children of n aggregated with the g packets generated at node n. For leaf nodes, $z^n_x = y_{x,g,a_n,l_n}$ with $x \leq g$, as only the generated packets can be transmitted.

$h_{C_n,q}$ is defined as the sum of all permutations of how q packets can arrive at node n from its children. For each permutation, we multiply the probabilities $z^{c_1}_{x_1}, z^{c_2}_{x_2}, \ldots, z^{c_{|C_n|}}_{x_{|C_n|}}$, where $C_n = \{c_1, c_2, \ldots, c_{|C_n|}\}$ and $q = \sum_{i=1}^{|C_n|} x_i$. For example, let us assume that node n has two children $C_n = \{c_1, c_2\}$, then $h_{C_n,q}$ for $q = 2$ is calculated as:

$$h_{C_n,2} = z^{c_1}_2 \cdot z^{c_2}_0 + z^{c_1}_0 \cdot z^{c_2}_2 + z^{c_1}_1 \cdot z^{c_2}_1 \tag{4}$$

which represents the probability that exactly 2 packets arrive from the children of node n.

The above calculations are performed in an iterative fashion for all nodes in the topology tree, starting from the leaf nodes, continuing with their parents, and so on, until the root is reached. At the root, we can calculate the expected total number of delivered packets as follows:

$$\sum_{q=0}^{|D_{root}| \cdot g} h_{C_{root},q} \cdot q \tag{5}$$

B. Radio On Time Calculation

Here we calculate the expected radio on time of the entire network during a single TSCH slot frame. To do so, we calculate the radio on time of each node separately and aggregate those to one total value. The expected number of packets a node will have in its queue at the beginning of the slot frame and the number of slots that will be used by the node and its parent to effectively transmit and receive these packets, is calculated. Using those values and the different radio on values for the different states a TSCH node can be in, the radio on time for the entire network can be approximated.

To calculate the radio on time for the packets transmitted from node n to its parent, we first approximate the expected number of packets that there will be in the queue of the node, e_n, defined as follows:

$$e_n = round\left(g + \sum_{q=0}^{|D_n| \cdot g} h_{C_n,q} \cdot q\right) \tag{6}$$

with g packets being generated at the node, $|D_n| \cdot g$ being the maximal number of packets that can arrive from the children of node n and $h_{C_n,q}$ as defined in Section IV-A2.

In every slot frame, a node n has a_n slots allocated towards its parent. Subsequently, we calculate the average number of slots that node n is expected to use, u_n, out of the a_n allocated slots, to transmit its queued packets e_n to its parent:

$$u_n = a_n - \sum_{v_i \in J_x} \frac{b^l_{v_{start},v_i}}{\sum_{v_i \in J_x} b^l_{v_{start},v_i}} \cdot a' \tag{7}$$

where a' is the number of slots not used by node n for every state $v_j = (q', a', r'_{max}, x)$, with q' being the packets left in the queue and r'_{max} being the number of unused retransmissions. $v_i \in J_x$ represents all the absorbing states with x arrived packets at the parent of node n, starting from state $v_{start} = (q, a, r_{max}, 0)$ over a link to its parent with reliability l.

Finally, we can calculate the radio on time for the communication between a node n and its parent as follows:

$$\begin{aligned} \sum_{x=0}^{min(a_n,e_n)} y_{x,e_n,a_n,l} \cdot (x \cdot (o_{txDataRxAck} + o_{rxDataTxAck}) + \\ (u_n - x) \cdot (1 - l) \cdot (o_{txData} + o_{rxIdle}) \quad + \\ (u_n - x) \cdot l \cdot (o_{txDataRxNack} + o_{rxDataTxNack}) \quad + \\ (a_n - u_n) \cdot o_{rxIdle}) \end{aligned} \tag{8}$$

with $y_{x,e_n,a_n,l}$ being the probability that at the end of the slot frame x packets have arrived at the parent, as described in Equation 2. The o_{state} values are the different radio on time values for the different states a TSCH node can be in, using the modulation chosen by node n. These different states are described by Daneels et al. in [23] and the radio on time value of each state is calculated similarly to the values in Table I for each modulation. The radio on time of every node in the network is calculated and summed, resulting in the radio on time of the entire network.

C. TSCH Slot Bonding Problem Formulation

Using the expected number of delivered packets and radio on time calculations as described in the previous sections, we propose the slot bonding problem formulation that maximizes the number of delivered packets while keeping the radio on time to a minimum. It is important to note that the problem formulation assigns slots in a single TSCH slot frame and this schedule is assumed to be repeated every slot frame.

1) Input Variables: The network consists of a set of nodes $N = \{n_0, n_1, ..., n_{max}\}$, of which node n_0 is the root, i.e., the sink node. $N_0 \subset N$ represents the set of nodes without the root n_0. $M = \{m_0, m_1, ..., m_{max}\}$ is the set of possible MCSs for each $n \in N_0$. The set of slots in a slot frame are denoted by $T = \{t_0, t_1, ..., t_{max}\}$ and the set of channel offsets at which a node can transmit are represented by $F = \{f_0, f_1, ..., f_{max}\}$. Every node n has a set $I_n \subset N_0$ which contains the set of nodes that interfere with the receiving node n, i.e., all the nodes that have a Received Signal Strength Indicator (RSSI) value at n higher than the channel noise floor, added to the noise figure of the receiver device. Each MCS $m \in M$, requires a bonded slot consisting of $s_m \in S$ regular slots, with $S = \{1, ..., |T|\}$, to transmit a 127 bytes packet and receiving an ACK.

2) Decision Variables: The first binary decision variable $\sigma_{t,f,s,n}$ represents the specific assigned bonded slots a node should select in the TSCH schedule. $\sigma_{t,f,s,n}$ is 1 when node $n \in N_0$ transmits in a bonded slot, using $s \in S$ consecutive regular slots allocated at time offsets $\{t, t+1, \ldots, t+(s-1)\} \subset T$ and channel offset $f \in F$ in the TSCH schedule matrix to its parent $p \in N$, else it is 0.

The second binary decision variable $\gamma_{p,m,n}$ is 1 when node $n \in N_0$ selects $p \in N$ as its parent and $m \in M$ as the modulation to transmit to the parent, else it is 0.

3) Objective: The objective is to maximize the expected number of delivered packets at the root in one slot frame, while keeping the radio on time to a minimum.

Therefore, the goal is to maximize Equation 5 that calculates the delivered packets at the root, given a TSCH network determined by the values assigned to the $\sigma_{t,f,s,n}$ and $\gamma_{p,m,n}$ variables. Simultaneously, for those solutions that provide the same number of delivered packets, the solution with the smallest radio on time of the network, defined as the sum of all radio on time values of all nodes in the network, should be preferred. To calculate the radio on time of the entire network, we need to calculate the radio on time for every node n using the values assigned to the $\sigma_{t,f,s,n}$ and $\gamma_{p,m,n}$ variables and Equation 8.

4) Constraints: A node $n \in N_0$ can choose only one parent $p \in N$ and one modulation $m \in M$ to transmit to that parent p:

$$\forall n \in N_0 : \sum_{p \in N} \sum_{m \in M} \gamma_{p,m,n} = 1 \tag{9}$$

A node $n \in N_0$ that uses MCS $m \in M$ to its parent $p \in N$, should allocate exactly s_m consecutive slots for a bonded slot transmission:

$$\forall n \in N_0, p \in N, t \in T, f \in F, m \in M, s \in S | s \neq s_m : \\ \gamma_{p,m,n} \cdot \sigma_{t,f,s,n} = 0 \tag{10}$$

A bonded slot transmission $\sigma_{t,f,s,n}$ consisting of s consecutive slots should not exceed the slot frame length:

$$\forall n \in N_0, t \in T, f \in F, s \in S : (t + s - 1) \cdot \sigma_{t,f,s,n} \leq t_{max} \tag{11}$$

A transmission of node $n \in N_0$ to its parent cannot overlap with any other transmission of the same node, nor with any of the transmissions of the nodes that have node n as their parent, as a node can only perform one action (i.e., transmit or receive) during each slot:

$$\forall n \in N_0, t \in T : \\ \left(\sum_{f \in F} \sum_{t' \in [0,t]} \sum_{s \in [t-t'+1,(t_{max}+1)-t']} \sigma_{t',f,s,n} \right) + \\ \left(\sum_{f \in F} \sum_{t' \in [0,t]} \sum_{s \in [t-t'+1,(t_{max}+1)-t']} \sum_{m \in M} \sum_{j \in N_0} \gamma_{n,m,j} \cdot \sigma_{t',f,s,j} \right) \\ \leq 1 \tag{12}$$

The transmissions of the children of the root cannot be in the same slot, as the root can only listen to one child at once:

$$\forall t \in T : \\ \sum_{f \in F} \sum_{t' \in [0,t]} \sum_{s \in [t-t'+1,(t_{max}+1)-t']} \sum_{m \in M} \sum_{j \in N_0} \gamma_{n_0,m,j} \cdot \sigma_{t',f,s,j} \\ \leq 1 \tag{13}$$

A transmission of node $n \in N_0$ to its parent p cannot overlap with any other transmission of the interferers of the receiving parent node p:

$$\forall n \in N_0, t \in T, f \in F : \\ \left(\sum_{t' \in [0,t]} \sum_{s \in [t-t'+1,(t_{max}+1)-t']} \sigma_{t',f,s,n} \right) \cdot \\ \left(\sum_{t' \in [0,t]} \sum_{s \in [t-t'+1,(t_{max}+1)-t']} \sum_{p \in N} \sum_{m \in M} \sum_{j \in I_p} \gamma_{p,m,n} \cdot \sigma_{t',f,s,j} \right) \\ = 0 \tag{14}$$

The output of this slot bonding formulation is a set of MCSs and bonded slot allocations from each node to its chosen parent, represented by the $\sigma_{t,f,s,n}$ and $\gamma_{p,m,n}$ decision variables, in order to maximize the expected number of delivered packets at the root while minimizing the radio on time. However, the proposed slot bonding problem becomes highly computationally complex when solving it for large topologies. The slot bonding formulation sub-problem of finding slot allocations for all nodes in a saturated network scenario is equivalent to the scheduling problem formulated by S.C. Ergen *et al.*, which is proven to be NP-complete [26]. Additionally, the sub-problem of building the topology tree by making parent and modulation selections can be seen as equivalent to the construction of the minimum routing cost spanning tree which is NP-hard [27]. As such, solving the slot bonding formulation is not possible in polynomial time.

V. A Heuristic Approach

In this section, detailed information is provided on the GA that allows us to implement and solve the slot bonding problem defined in Section IV. First, we describe the genetic algorithm and its different operators. Afterwards, the feasibility heuristic that is used to know if a GA solution fits in the TSCH schedule, is explained in detail.

A. Genetic Algorithm

A GA is a heuristic technique to solve optimization or search problems by simulating natural evolution [28]. It is an iterative process that starts from an initial randomly or heuristically created population. A population consists of so-called individuals that represent candidate solutions of the problem that is being solved. Each generation (i.e., iteration), the GA applies biologically inspired operators to select appropriate individuals and do cross-over and mutation operations on the individuals in the population, which results in a new offspring population. This offspring combined with the population at the start of the generation will be the population for the next generation in the evolutionary process. To select appropriate individuals for the operators and the next generations, each individual is evaluated and assigned a fitness value during each generation. This value represents its performance measured by the optimization problem's objective function. The goal of a GA is to evolve towards a population that contains the best candidate solutions for the problem at hand.

We applied a GA to find the best candidate solutions for the slot bonding problem defined in Section IV. Most importantly, the operators (see Sections V-A2 and V-A3) and objective function (see Section V-A4) of the GA can easily be adjusted to, respectively, respect the validity of the network topology and make the complex delivered packets calculations. Additionally, the choice of a parent, MCS, and the number of allocated TSCH slots can easily be encoded in a GA integer string (see Section V-A1), similarly to what was argued by M. Ojo *et al.* [29]. Also, as GAs are well-fitted for finding global solutions for problems with large search spaces and the search space of our slot bonding problem is huge, applying a GA for this problem is suitable. For example, for a network with only 5 nodes except the root, so 5 potential parents, 2 potential MCSs and a slot frame length of 9 slots (so 10 possible numbers of allocated slots), the size of the solution space is already $(5 \cdot 2 \cdot 10)^5 = 10^{10}$.

$$\underbrace{p_1\ m_1\ a_1}_{\text{node 1}}\ \underbrace{p_2\ m_2\ a_2}_{\text{node 2}}\ \underbrace{p_3\ m_3\ a_3}_{\text{node 3}}\ \dots\ \underbrace{p_{n-1}\ m_{n-1}\ a_{n-1}}_{\text{node n-1}}$$

Fig. 3. GA individual for a topology of n nodes with 3 genes per node. The root is not explicitly represented in the individual.

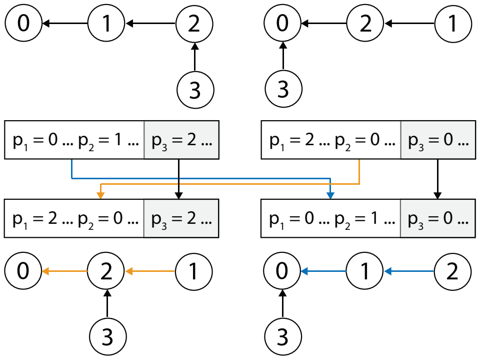

0
1
2
3
0
2
1
3
p1 = 0 ... p2 = 1 ... | p3 = 2 ...
p1 = 2 ... p2 = 0 ... | p3 = 0 ...
p1 = 2 ... p2 = 0 ... | p3 = 2 ...
p1 = 0 ... p2 = 1 ... | p3 = 0 ...
0
2
1
3
0
1
2
3

Fig. 4. Example of successful two-point cross-over operation where the genetic information of the first two nodes is interchanged, resulting in two new individuals with valid topologies.

The inputs to the GA are the set of network nodes N, the set of interferers I_n for each node n, the set of different available MCSs M that all result in a specific reliability l for each node-parent pair, the length of the bonded slots for each modulation $s_m \in \mathbb{N}_{>0}$, the radio on time o_{state} for each state a TSCH node can be in and for every modulation m that can be chosen, and the TSCH schedule with its set of slots T and frequency channels F. The output of the GA is a parent selection for each node in the network and MCS and slot allocations for each node to its chosen parent, in order to maximize the expected number of delivered packets at the root while minimizing the radio on time. In this section, we explain the candidate solution representation, the different operators, the fitness function, and provide an exact overview of the GA workflow.

1) Representation: A candidate solution of the slot bonding problem is represented by a GA individual as shown in Figure 3. Each node in the individual is represented by 3 so-called *genes*, being an encoder of a characteristic of the node, with each gene value being a natural number. The first gene p_n represents the parent of the node n, m_n represents the MCS that node n applies to communicate with its parent and the last gene a_n is the number of bonded slots it allocates to its parent. The root node does not need representation in the individual as other nodes connect and allocate resources to the root.

2) Cross-over Operator: A cross-over operation recombines the genetic information of two individuals into new offspring individuals. A slightly altered version of the traditional two-point cross-over was applied. The two-point cross-over was chosen over the single-point crossover to reduce the positional bias [30]. Traditionally, this technique chooses two random positions in the individuals, after which both parent individuals are cut at those positions and the genetic substring

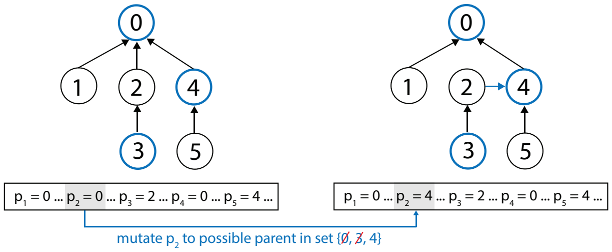

0
1
2
4
3
5
p₁ = 0 ... p₂ = 0 ... p₃ = 2 ... p₄ = 0 ... p₅ = 4 ...
0
1
2
4
3
5
p₁ = 0 ... p₂ = 4 ... p₃ = 2 ... p₄ = 0 ... p₅ = 4 ...
mutate p₂ to possible parent in set {~~0~~, ~~3~~, 4}

Fig. 5. Example of the mutation operation in which the parent of node 2 is altered. Assuming that the possible parent set of node 2 is $\{0, 3, 4\}$, the new parent can only be node 4, as the current parent is 0 and node 3 is a descendant of node 2 (which would lead to a loop).

between those positions of the first parent is interchanged with the substring at the same location in the second parent. We applied two changes to that operator implementation. First, the random positions are limited to be multiples of three in order to never separate the three genes of one node. Second, if interchanging the substring between the chosen positions does not result in a new valid topology tree (i.e., there is a path from each node to the root and there is no path loop), we decrement the substring with 3 genes (i.e., 1 node) and retry if this cross-over results in a valid tree. This process is repeated until this results in a valid cross-over or until the length of substring is 0 and thus no cross-over is applied. Figure 4 shows a successful crossover operation in which the genetic information of the first two nodes is interchanged, resulting into 2 new individuals with valid topologies.

3) Mutation Operator: A mutation operator introduces genetic diversity into the population by randomly altering one or more genes of the individual. We implemented a 3-phase mutation operator. In the first phase, our mutation operator iterates over all nodes in the individual and with a probability p_{gene} alters the parent gene p_n of a node n. This parent is chosen out of the list of valid parents for that node, i.e., based on the topology information and the communication range of the different available MCSs. From this list of possible parents, all nodes that are currently descendants of the node are excluded to prevent the mutation operator from introducing a loop in the network tree. The current parent is also excluded. Figure 5 shows an example of such a successful parent mutation. In the second phase, the mutation operator iterates over all MCS genes of all nodes in the individual and, with the same probability p_{gene} or if the parent of this node was changed, randomly alters the gene value m_n. When picking a new MCS, the possible MCSs depend on the current parent of the node to prevent it from choosing a modulation to its parent that results in reliability that is below the predefined threshold to form a link (i.e., see the experiment setup in Section VI-A for the threshold value). In the third and last phase, the operator iterates over the third gene of all nodes, i.e., the number of bonded slots a node allocates a_n. It alters the gene value with the probability p_{gene} or whenever the MCS value of a node was altered (to make sure the new number of slots value does not exceed the slot frame length). The valid range for the number of slots depends on the picked MCS, and is $[0, \left\lfloor \frac{len(slotframe)}{len(bonded\_slot, MCS)} \right\rfloor]$ in which $len(bonded\_slot, MCS)$ represents the number of regular slots needed to bond together for the chosen MCS.

Algorithm 1 Genetic algorithm workflow

```
best_ind ← None
generation ← 0
pop ← initialize_population(pop_size)
evaluate_fitness(pop)
while generation ≤ max_generations do
    parent_pop ← select(pop, pop_size)
    offspring ← crossover(parent_pop)
    offspring ← mutate(offspring, $p_{gene}$)
    evaluate_fitness(offspring)
    pop ← select(pop + offspring, pop_size)
    for ind ∈ pop do
        if ind.fitness > best_ind.fitness then
            best_ind ← ind
return best_ind
```

4) Fitness function: To evaluate the fitness of an individual according to the slot bonding problem formulated in Section IV, the fitness function calculates the expected total number of delivered packets at the root and the network radio on time. To calculate the total number of delivered packets, we apply the calculation defined in Section IV-A. The radio on time for each node is calculated by applying Equation 8. These radio on times are summed, resulting in the network radio on time. Consequently, the individual with the highest expected number of delivered packets and lowest radio on time, in lexicographical order, is considered the best individual. This means that the first goal of the GA is maximizing the expected number of delivered packets and then individuals with identical delivered packets values are ranked according to their radio on time. However, before evaluating the expected number of delivered packets and radio on time, an individual is first checked on being *feasible*, meaning that all allocations should fit the in TSCH schedule. To do so, the individual is passed to the feasibility checking algorithm defined in Section V-B. If the individual is not feasible, the number of delivered packets is set to a negative value (i.e., -100) and the radio on time to an unrealistic high positive value (i.e., 1×10^6 ms). In case the individual is feasible, the fitness function continues to calculate the expected number of delivered packets and the radio on time.

5) Workflow: Algorithm 1 shows the workflow of the genetic algorithm. Before the GA starts for a run of $max\_generations$ generations, the population pop is initialized with pop_size number of individuals. The initial individuals are built by starting from one and the same valid topology (i.e., this initial topology is the same for all individuals in the initial population), while the m_n and a_n genes are picked uniformly at random out of the set of valid options for that node n to its parent, meaning that only MCSs can be chosen that allow the node to communicate with its parent and the length of all allocated bonded slots does not exceed the slot frame length. Then, the mutation operation of Section V-A3 is applied 100 times to randomize the individual while guaranteeing that the MCS allocations,

Algorithm 2 Feasibility heuristic

```
 1: for n ∈ $N_0$ do
 2:     if $a_n > 0$ then
 3:         num_slots = $a_n$
 4:         for $f \in [0, f_{max}]$ do
 5:             for $t \in [0, t_{max} - s_{m_n} + 1]$ do
 6:                 if allocate($n, s_{m_n}, t, f$) then
 7:                     num_slots = num_slots $-$ 1
 8:                     if num_slots $> 0$ then
 9:                         continue with next slot
10:                     else
11:                         continue with next node
12:         if num_slots $> 0$ then
13:             return False
14: return True
```

the slot allocations and the represented topology stay valid (i.e., each node can reach the root and there are no routing loops). In each generation, on line 6, it first selects pop_size parents for reproduction. Then, the cross-over operator as discussed in Section V-A2 is applied to all sets of two parents in $parent\_pop$, after which the GA mutates each individual of the $offspring$, with a probability p_{gene} being the probability to alter a gene of the individual, according to the operator described in Section V-A3. Afterwards, at line 9, the fitness of the $offspring$ is evaluated using the fitness function of Section V-A4. The population for the next generation is prepared by selecting pop_size individuals out of the population pop that started the generation and the produced $offspring$. At lines 11-14, the fitness of each individual in the new population is checked to be better than the fitness of the current best individual $best\_ind$. If so, that individual becomes the new $best\_ind$. This procedure is repeated until the $max\_generations$ is reached, after which the $best\_ind$ is returned.

The exact probability p_{gene}, and the different selection strategies of line 6 and 10, were selected empirically and are discussed later in Section VI-A.

B. Feasibility Heuristic

An individual returned by the proposed GA should be feasible, meaning that all allocations should fit in one TSCH slot frame while avoiding interference and respecting that a node cannot transmit and receive simultaneously. To check if this holds for the given individual, it is possible to formulate a set of linear constraints that the solution should satisfy, based on the formulation in Section IV-C. However, because of the computational complexity of such a feasibility check with linear constraints, instead we employ a greedy bin-packing heuristic that guarantees to identify all infeasible solutions (while it might identify a small percentage of the feasible solutions as infeasible). The heuristic takes as input the network tree topology, the number of bonded slot allocations a_n for the chosen MCS between a node n and its parent, as represented in the tested individual, and the interferers of all nodes in the network. This heuristic is also used to generate a possible TSCH schedule from the best individual found by the proposed genetic algorithm.

The heuristic is shown in Algorithm 2. It iterates over all nodes except the root, N_0, and checks if the number of allocated bonded slots a_n of node n is larger than zero. If not, the heuristic can directly continue to the next node. The for loops of lines 4 and 5 iterate over the entire schedule and the heuristic tries to allocate the bonded slot of length s_{m_n} (with m_n being the modulation picked by node n) at each slot (t, f). To allocate a bonded slot starting at a slot (t, f), it checks if the node or the parent of the node is not yet transmitting or receiving in the range $[(t, f), (t + s_{m_n} - 1, f)]$ and if no slot in that range is used by an interferer. If the bonded slot can be allocated successfully, the heuristic continues with the next bonded slot for that node or with the next node, depending on if there are bonded slots left to allocate, i.e., $num\_slots > 0$ at line 8. If it cannot be allocated, the heuristic moves to the next slot. If the heuristic has iterated over all slots in the schedule and there are still bonded slots to allocate, i.e., $num\_slots > 0$ at line 12, the individual is considered infeasible. If the heuristic can iterate over all nodes and allocate all bonded slots, the individual is feasible.

It is important to stress that the bin-packing heuristic will correctly identify *all* infeasible individuals. However, depending on the order of the nodes in N_0, some feasible individuals will be identified as infeasible. Therefore, for each individual, there are different runs of the heuristic in Algorithm 2, each time with N_0 sorted differently, such as sorting the nodes with most bonded slot allocations first or using a breadth-first search of the topology tree. As long as the heuristic labels the individual as infeasible and there are sorting algorithms left to try, the heuristic runs again. From the moment an individual is considered feasible by the heuristic, this process stops and the positive outcome is returned. If none of the sorting algorithms make the heuristic return a feasible result, the individual is returned as an infeasible solution.

C. Time Complexity Analysis

To analyse the time complexity of the GA, the different steps of a single generation in Algorithm 1 (i.e., the while-loop from line 5 to 13) were analysed. The time complexity of the cross-over operator (i.e., $O(|N|^3)$), the mutation operator (i.e., $O(|N|^2)$), the selection operator at line 6 (i.e., tournament selection with time complexity $O(k \cdot pop\_size)$ and k being the tournament size), the selection operator at line 10 (i.e., elitist selection with time complexity $O(pop\_size \cdot log(pop\_size))$) are all dominated by the time complexity of the fitness function calculation (specifically, the calculation of the $h_{C_n,q}$ values at every node, as defined in Section IV-A2), resulting in a total time complexity of $O(|N|! \cdot |N|^2)$, while assuming fixed constants for parameters such as $max\_generations, pop\_size, Q, |F|, |T|$ and g. The analysis confirms the high complexity of solving the proposed slot bonding formulation.

VI. Evaluation

In this section, we evaluate the effectiveness of the decisions taken by the proposed GA, in terms of the parent selections and MCS and slot allocations, to solve the proposed TSCH slot bonding problem by applying them in TSCH network simulations. First, we compare the GA heuristic to the optimal solution obtained from an exhaustive search approach. Afterwards, we test the effectiveness of the feasibility heuristic proposed in Section V-B and validate the proposed GA by comparing its results to the simulation results. Then, we run the GA for different network sizes while showing the advantages of slot bonding. We also show the advantage of employing multiple modulations, compared with only using one modulation. Finally, we provide insight in preferred parent selection and modulation configuration when applying slot bonding that can be used as a basis to develop practical scheduling and routing algorithms for multi-MCS TSCH networks.

A. Experiment Methodology & Setup

The experiments are conducted by following a four-step process: first, the network topologies were generated by the 6TiSCH simulator [31]. Second, those topologies were fed to the GA, implemented in the Python DEAP framework [32], that solves the slot bonding problem defined in Section IV heuristically. Third, the best possible solution found by the GA was used as input for the heuristic defined in Section V-B to generate a TSCH schedule for all nodes in the network. Fourth, the scheduling solution provided by the heuristic is used as a centralized schedule to make static schedule allocations in the TSCH experiments using the 6TiSCH simulator. This evaluation process that allows us to analyse the slot bonding technique is illustrated in Figure 6. The source code changes to the 6TiSCH simulator, the GA implementation and the feasibility heuristic are publicly available[1]. The number of slots per bonded slot are configured as shown in Table I. Every node of the network is configured to generate 1 packet per slot frame. Each experiment result is averaged over at least 20 iterations. The error bars in the figures represent the standard deviation over these iterations.

1) 6TiSCH Simulator: We extended the simulator with a sub-GHz outdoor propagation model based on the 3rd Generation Partnership Project (3GPP) spatial channel model [33] and use 3 channels in the 868 - 868.6 MHz band. The simulator supports the 3 different modulations in option 4 of the OFDM PHY which all require $156\,\mathrm{kHz}$ nominal bandwidth per channel, as listed in Table I. In addition, to determine the link reliabilities for every modulation (as introduced in Section IV-A2), real-world measurements were performed with the Atmel AT86RF215 transceivers[2] (which is fully compliant with the IEEE 802.15.4g-2012 standard and thus supports OFDM) of two OpenMote-B nodes. After configuring both nodes with the same modulation scheme, attenuating the transmitter by a certain value and transmitting 200 frames with a 127 byte Physical Service Data Unit (PSDU), the packet reception rate (PRR) and average RSSI were calculated from the receiver logs. This process was repeated for each combination of modulation scheme and attenuation in order to map the PRR in function of average RSSI for each of the selected modulation schemes. The measured data and resulting regression models are publicly available[3].

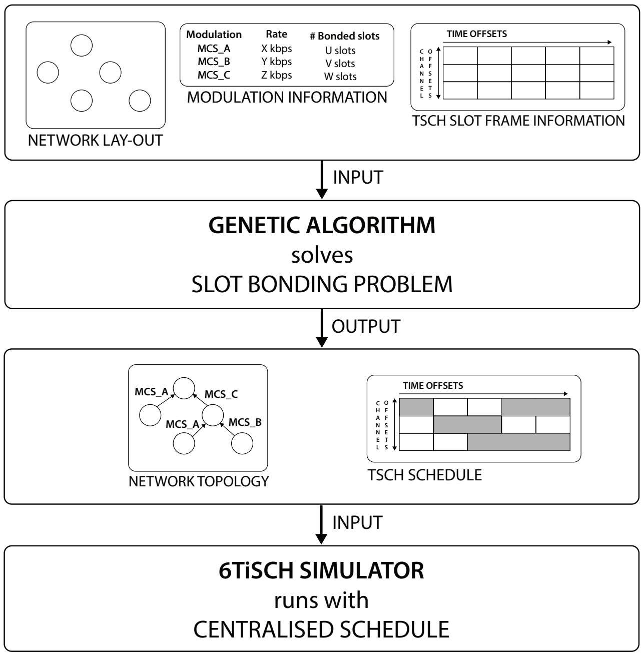


Fig. 6. Flow diagram showing the evaluation process, showing that the output of the GA is used as a centralised schedule for the 6TiSCH simulator to analyse the proposed slot bonding technique.

The topologies were created by randomly placing nodes until a node had at least one reliable link (i.e., a PDR of at least 70% for MCS2) to one of the other nodes. This was done to ensure that the network is fully connected and each node can reach the sink over one or multiple hops. In our experiments, only the PHY and MAC layer were enabled and all other stack layers were disabled. The interference model implemented in the simulator, was changed to the Yet Another Network Simulator (YANS) model [34]. Note that while the interference model was enabled during all simulations, there were no transmission failures due to interference effects because the proposed slot bonding model includes interference avoidance. The receiver noise floor is calculated using the thermal noise calculation $N = 1.381 \cdot 10^{-23} \frac{J}{K} \cdot 290K \cdot 156 \cdot 10^{3}\mathrm{Hz}$, added to the noise figure of the Atmel AT86RF215 chip, i.e., $4.5\,\mathrm{dB}$.

2) GA: The GA workflow is configured as described in Section V-A. The population size pop_size is set to 100 individuals, while the maximum number of generations $max\_generations$ is configured to 10000 generations. The parent selection operator, at line 6 in Algorithm 1, is set to tournament selection with a tournament size of 2. The mutation probability of a gene p_{gene} is set to 0.05. Within the mutation operator, the minimum reliability threshold for a

[1] https://github.com/imec-idlab/adaptive-mcs-ga-simulator

[2] http://ww1.microchip.com/downloads/en/devicedoc/atmel-42415-wireless-at86rf215_datasheet.pdf

[3] https://github.com/imec-idlab/adaptive-mcs-ofdm-measurements

node to be able to pick a specific modulation to its parent, is 70 %. To select the population for the next generation, at line 10 in Algorithm 1, elitist selection, retaining the 10 % best individuals of the previous generation, is applied. This GA configuration showed the best overall performance after evaluating a large number of configurations for a wide variety of inputs (results omitted due to space constraints). To calculate the expected number of delivered packets and the network radio on time in the fitness function of the GA as described in Section IV-A and Section IV-B, the output of the topology creation is provided as input to the GA, i.e., the different available MCSs and corresponding link reliabilities between all network nodes. The parameter g is set to 1 as each node is configured to generate 1 packet per slot frame. The maximum number of allowed transmission opportunities per packet, denoted by r_{max} in Section IV-A1, is set to 4 as advised by the 6TiSCH Minimal Configuration [35].

B. GA Validation

We compare the expected number of delivered packets and radio on time results returned by the GA to the results returned by exhaustive search. Due to the computational complexity, we could only run exhaustive search experiments for topologies up to 7 nodes for 10 ms regular slots (while applying slot bonding) and 8 nodes for 40 ms regular slots. The TSCH schedule was configured to have a slot frame length of 120 ms, 2 channels and 2 modulations, i.e., MCS2 and MCS4 of Table I. Next to the optimality of delivered packets and the difference in radio on time between the GA and the exhaustive search, Table III also shows the relative number of candidate solutions (i.e. individuals) checked by the GA and the relative computation time needed by the GA compared with the exhaustive search. The relative number of candidate solutions is the number of unique solutions checked by the GA over the number of candidate solutions checked by the exhaustive search (after filtering out a small set of invalid solutions beforehand). The relative computation time is the time needed by the GA over the time needed by the exhaustive algorithm to find the optimal solution. As expected, for an increasing number of nodes and the shorter the regular slot length (which results in more slots in a slot frame), the relative number of candidate solutions and relative computation time for the GA becomes smaller as the number of candidate solutions increases exponentially. For 5 nodes and a regular slot length of 40 ms, we notice that the GA took slightly more time to complete than the exhaustive search while considering less unique candidates. The reason is that the GA employs a fixed population size (i.e., 100 individuals) and a fixed number of generations (i.e., 10000 generations), resulting in a total of 1000000 solutions that need to be checked by the GA. As for this experiment configuration the number of unique solutions is smaller than 1000000, the total number of individuals checked by the GA includes many duplicate solutions that require extra computation time. The delivered packets and radio on time differences in Table III show that, for topologies up to 5 nodes and a regular slot length of 10 ms, the GA finds a globally optimal solution, while for 6 and 7 the optimality drops with 0.6 % and 0.4 % respectively. For a regular time slot length of 40 ms a globally optimal solution is found for all topology sizes. These results confirm that the proposed GA is capable of finding (near-)optimal solutions for the slot bonding problem.

TABLE III
AVERAGED GA RESULTS COMPARED WITH EXHAUSTIVE SEARCH RESULTS, FOR DIFFERENT TOPOLOGY SIZES AND SLOT LENGTHS.

Nr. nodes, slot length	Delivered packets optimality (%)	Radio on difference (ms)	Relative nr. candidates (%)	Relative comp. time (%)
5 nodes, 10 ms	100	0	0.6	21.5
6 nodes, 10 ms	99.4	1.7	< 0.1	3.1
7 nodes, 10 ms	99.6	5.2	< 0.1	0.5
5 nodes, 40 ms	100	0	3.7	102
6 nodes, 40 ms	100	0	0.3	8.8
7 nodes, 40 ms	100	0	< 0.1	1.1
8 nodes, 40 ms	100	0	< 0.1	0.6

TABLE IV
FEASIBILITY HEURISTIC COMPARED WITH THE EXACT INTEGER LINEAR PROGRAM (ILP) FEASIBILITY MODEL FOR A SLOT FRAME LENGTH OF 200 ms AND 2000 GENERATIONS.

Topology size	Relative comp. time (%)	True negative rate	False negative rate
8	3.9 ± 0.5	1.0 ± 0	0.17 ± 0.014
14	2.4 ± 0.7	1.0 ± 0	0.058 ± 0.039

To show the accuracy of the feasibility heuristic presented in Section V-B, the output of the heuristic is compared with the output of a computationally-intensive Integer Linear Program (ILP) model that can determine the feasibility of a GA individual with 100 % accuracy. This ILP is extracted from the set of decision variables and constraints defined in the slot bonding formulation in Section IV. Table IV shows the results for a slot frame length of 200 ms and 2000 generations. The heuristic proves to be much faster as it needs only 3.9 % and 2.4 % of the ILP computation time. The results also show that the heuristic identifies all infeasible individuals, i.e., the true negative rates are 1. However, the heuristic does label some individuals as infeasible while they are feasible: the false negative rate of 0.017 for 8 nodes increases to 0.058 for 14 nodes. While the increasing number of nodes clearly makes it more difficult for the heuristic to fit all allocations in the TSCH schedule, this has a minimal impact on the expected delivered packets results. When comparing the best solution found when using the heuristic with the best solution found when using the ILP, the average optimality of the number delivered packets is 99.2 % ± 2.6 % and 98.0 % ± 7.4 % for 8 and 14 motes respectively.

We also validate the GA by comparing its PDR results (i.e., the expected delivered packets value calculated in Equation 5, over the number of generated packets per slot frame) with the PDR results of the TSCH simulation that used the GA

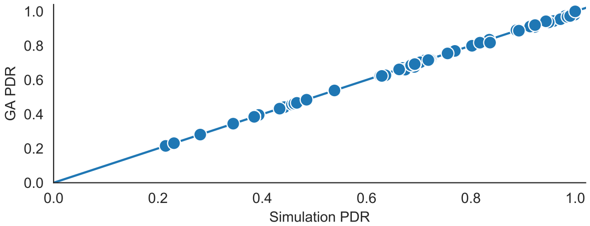


Fig. 7. GA and simulation PDR for all experiment iterations for the 10 ms bonded slots for 120 ms, 200 ms, 280 ms and 360 ms slot frame lengths in a network with 14 nodes.

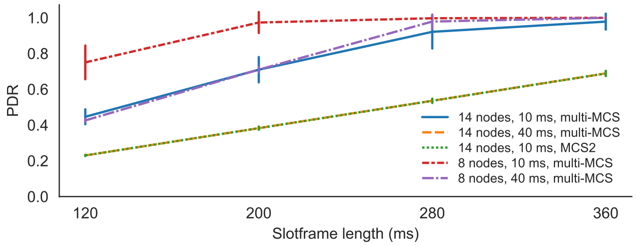


Fig. 8. PDR comparison for 10 ms bonded slots and 40 ms slots, for different slot frame lengths.

schedule allocations as input. Figure 7 compares the GA and simulation PDR values for all iterations for the 10 ms bonded slots and 40 ms experiments with 120 ms, 200 ms, 280 ms and 360 ms slot frame lengths in a network with 14 nodes. For all iterations, both values are nearly identical with an overall root mean squared error (RSME) of only 0.0044, confirming that the GA model and TSCH simulator behave similar and validating that the expected PDR calculated by our GA model is accurate.

C. Slot Bonding Scalability

To show the effect of slot bonding on scalability, we compare the resource-efficient slot bonding technique with the static approach without slot bonding where a time slot should have a length long enough to accommodate all possible MCSs (and thus valuable slot frame time is wasted for transmissions using MCSs with fast data rates that do not require the maximum slot length). We compare TSCH schedules that contain 10 ms bonded slots to schedules with 40 ms regular slots. Table I shows the number of slots needed for each supported MCS. The 40 ms slots are long enough to support all MCSs and thus no slot bonding is required. We compare both techniques for slot frame lengths of 120 ms, 200 ms, 280 ms and 360 ms, meaning that there are 12, 20, 28 and 36 slots for the slot frame with 10 ms bonded slots and 3, 5, 7 and 9 regular slots for the slot frame with 40 ms slots respectively.

Figure 8 shows the average PDR GA results for 8 and 14 nodes. It is clear that the slot bonding technique offers a clear PDR advantage over the static approach for different slot frame lengths. For a 120 ms slot frame length and 8 nodes, the PDR increases from 0.426 (± 0.011) to 0.75 (± 0.095), while for 14 nodes the PDR almost doubles from 0.23 (± 0.004) to 0.446 (± 0.042). For the slot frame length of 360 ms, the difference for 8 nodes between both PDRs is negligible as both PDRs are close to 1 because the slot frame length is long enough to accommodate all the necessary bonded slots. Additionally, applying slot bonding allows to achieve a PDR near 1 at a 200 ms slot frame length, while for 40 ms this requires a 280 ms length. For 14 nodes and a slot frame length of 360 ms, there is still a PDR increase of 42 % with the 40 ms approach having a PDR of 0.689 (± 0.014) while the slot bonding approach has a PDR of 0.978 (± 0.045). As expected, for both techniques the PDR increases as the length of the slot frame increases and the contention decreases.

TABLE V
GA AVERAGED RESULTS FOR 14 NODE TOPOLOGIES, FOR BOTH 10 ms (WITH SLOT BONDING) AND 40 ms REGULAR SLOT LENGTHS AND DIFFERENT SLOT FRAME LENGTHS.

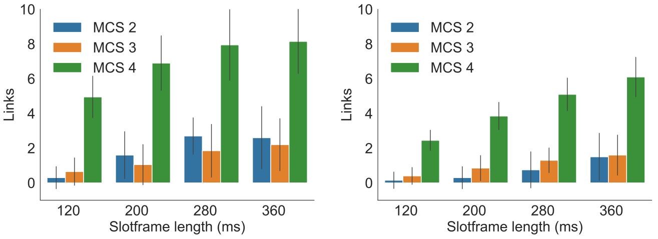
Slot length, Slot frame (ms)	Retransmission drops (%)	Full queue drops (%)	Link reliability (%)	Hop count
10, 120	0 ± 0	55.4 ± 4.2	99.1 ± 3.8	1.6 ± 0.6
10, 200	0.1 ± 0.1	29 ± 7.1	98.1 ± 5.5	1.9 ± 0.8
10, 280	0.5 ± 1.7	7.3 ± 9.2	97.6 ± 5.8	2.2 ± 1
10, 360	0.1 ± 0.1	2.1 ± 4.5	98 ± 5.5	2.2 ± 1
40, 120	0 ± 0	77 ± 0.4	99.7 ± 2.3	1.1 ± 0.3
40, 200	0 ± 0	61.7 ± 0.9	99.8 ± 1.9	1.4 ± 0.6
40, 280	0.2 ± 0.6	46.2 ± 0.9	99.5 ± 3.2	1.6 ± 0.7
40, 360	0.6 ± 1.8	30.5 ± 2.1	99.3 ± 3.4	1.8 ± 0.8

Table V shows the average number of drops due to the maximum of retransmissions being reached (i.e., caused by link failures or full queues at the receiver) relative to the number of generated packets, the average number of generated packets that was immediately dropped (because the queue of the transmitter was full), the average link reliability and the average hop count. For the last two metrics, only the nodes for which a packet could reach the root were considered, meaning only nodes for which there was a path to the root with every node along that path having at least one allocated slot to its parent. Because the reliability of the allocated links in both approaches was high, it is clear that the major reason for the difference in PDR is due to saturated queues, either at the transmitter or at the receiver.

These results show that the slot bonding technique is more efficient and can choose to allocate an extra slot with an MCS that requires less time (a bonded slot of 20 ms or 30 ms) which could be just short enough and have enough reliability to serve an extra node or packet in the network. In contrast, for each extra slot the 40 ms static approach wants to allocate, the complete slot length needs to be allocated, i.e., 40 ms. This confirms the scalability advantage of the slot bonding approach over the static approach with fixed 40 ms slot lengths.

D. Adaptive Modulations

We show the advantage of employing multiple modulations to adapt to the link conditions compared with only using a single modulation. Figure 8 shows the results in a network with



(a) $10\,\mathrm{ms}$ bonded slots

(b) $40\,\mathrm{ms}$ bonded slots

Fig. 9. Modulations used in a TSCH network with a topology size of 14 nodes.

14 nodes when applying the slot bonding approach with $10\,\mathrm{ms}$ slots for both multiple MCSs and only employing MCS2. Only employing MCS4 was not possible, as it cannot always create a reliable enough link to create a fully connected network. It is clear that using multiple modulations combined with slot bonding results in significantly better PDR values. The figure also shows that the PDR performance of only using MCS2 is the same as the performance of a network that only uses $40\,\mathrm{ms}$ regular slots (i.e., without slot bonding) for multiple modulations. The reason for this is that in order to use MCS2, 4 regular slots always have to be bonded together. However, when comparing the radio on time of both experiments, there is a clear difference in the experiment with multiple modulations as the nodes can choose modulations that may provide the same reliability as MCS2 but require less transmission time. For the slot frame lengths of $120\,\mathrm{ms}$, $200\,\mathrm{ms}$, $280\,\mathrm{ms}$ and $360\,\mathrm{ms}$, there was a respective $56.2\,\%$, $50\,\%$, $46.2\,\%$ and $43.7\,\%$ decrease in radio on time when using multiple modulations.

E. Allocation Analysis

First, to understand how the network tries to optimize the PDR, we observe the different modulations used by the nodes. Figure 9 shows the results for 14 nodes, when applying slot bonding with $10\,\mathrm{ms}$ and with $40\,\mathrm{ms}$ regular slots. Table V shows the average reliabilities for all the links in the network. The figure only shows the modulations for links of nodes for which a packet could reach the root. It is clear that when applying slot bonding more nodes in the network can make slot allocations, especially for the short slot frame lengths of $120\,\mathrm{ms}$ and $200\,\mathrm{ms}$, compared with the $40\,\mathrm{ms}$ naive approach. In general, the nodes try to allocate many links using MCS4, which is the fastest modulation and thus requires less transmission time. In the case of slot bonding, this means that only 2 regular slots are bonded together, thus leaving more free space in the schedule for other allocations. Moreover in the slot bonding case, when increasing the slot frame length, also more MCS2 modulations are allocated in an effort to guarantee more reliability as there is more free space in the schedule to do so.

Additionally, different metrics were tested to thoroughly understand the decisions that were taken when choosing a node's parent and modulation in order to optimize the PDR and radio on time. Table VI shows the different metrics and the average values over the results obtained for the $120\,\mathrm{ms}$, $200\,\mathrm{ms}$, $280\,\mathrm{ms}$ and $360\,\mathrm{ms}$ slot frame lengths for two GA experiments with respectively 8 and 14 nodes, when applying slot bonding with a $10\,\mathrm{ms}$ regular slot length. The results in the table show that when a node chooses its parent and the modulation to that parent, it first focuses on modulations that provide high reliability rather than preferring fast modulations, as the numbers of picking reliable modulations are higher than those of picking faster modulations. These results, combined with the picked modulations shown in Figure 9 and the reliabilities listed in Table V make it clear that many nodes were able to pick the fastest modulation MCS4 while still achieving high reliability. Also, when there are different modulations available at the parent that provide an equal reliability, the results show that in virtually all cases the node will pick the fastest modulation.

When looking at parent selection choices, parent nodes that are physically closer to the root than the node picking the parent, were clearly preferred with averages of $96.5\,\%$ and $94.3\,\%$ for 8 and 14 nodes respectively. Due to shorter distance, those neighbors experience less path loss than nodes that are further away from the root, making them the more reliable choice. More reliability means less transmissions needed to deliver the packets and thus less radio on time and slots that need to be allocated, again leaving more free space in the schedule for other nodes. Additionally, the nodes seemed to have a strong preference for parent nodes that had a reliability towards their respective parent that was equal or higher than the reliability between the node itself and its parent. On average, in $64.7\,\%$ and $81.8\,\%$ of the cases for the experiments with 8 and 14 nodes respectively, a node also preferred a parent node with equally fast or faster modulations to their parent compared to the rate of the modulation between the node itself and its parent. Moreover when only looking at nodes that allocated the slowest MCS2 modulation towards their parent, it is significant that on average in $95.3\,\%$ and $91.1\,\%$ of the nodes (i.e., in the experiments with 8 and 14 nodes respectively) prefer a parent that allocated a modulation with a faster rate than the modulation between the node itself and its parent. When looking at nodes with the fastest modulation MCS4 allocated towards their parent, we observe that especially for the shorter slot frame lengths nodes prefer parents with the same rate (it is not possible to allocate a faster modulation), while for the longer slot frame lengths this effect diminishes, i.e., for a $120\,\mathrm{ms}$ (and a network of 14 nodes) $91.3\,\%$ of the nodes prefer the same modulation, while for $360\,\mathrm{ms}$ length this is only $54.4\,\%$. As there is more space in the schedule with longer slot frame lengths, nodes probably start to prefer the slower but more reliable MCS2 modulations.

VII. Conclusion

This work explored the simultaneous use of multiple modulations in a IEEE 802.15.4e TSCH network by proposing and analysing the concept of slot bonding. We formally described the slot bonding problem that maximizes the expected number of delivered packets to the root while minimizing the radio on time. Additionally, we proposed a GA as a static analysis

TABLE VI
DIFFERENT METRICS TESTED FOR TWO GA RUNS FOR RESPECTIVELY 8 AND 14 NODES WHEN APPLYING SLOT BONDING WITH 10 ms, SHOWING THE AVERAGE VALUES OVER THE RESULTS OBTAINED FOR THE 120 ms, 200 ms, 280 ms AND 360 ms SLOT FRAME LENGTHS.

Metric	8 nodes	14 nodes
Picked most reliable modulation of parent / overall (%)	94.8 ± 7.8 85.3 ± 9.1	95.3 ± 1.6 88.5 ± 4
Picked fastest modulation of parent / overall (%)	70.5 ± 2.7 64.8 ± 2.7	77.2 ± 6.6 72.1 ± 9.1
Picked fastest modulation of equal reliable modulations of parent / overall (%)	99.9 ± 0.2 94.2 ± 0.9	100 ± 0 95.6 ± 1.9
Picked parent closer to root than itself (%)	96.5 ± 1.2	94.3 ± 2.8
Picked parent with equal or better reliability than itself (%)	77.5 ± 6.3	78.8 ± 7.2
Picked parent with equally fast or faster rate than itself (%)	64.7 ± 14.7	81.8 ± 9.8

tool, allowing us to implement and solve the computationally complex slot bonding problem up to a network size of 14 nodes. The GA finds an near-optimal solution in terms of parent and MCS selection as well as slot allocations, i.e., the maximum deviation in terms of delivered packets to the root is on average less than 1%.

The results of applying the GA solutions in TSCH network simulations confirmed the scalability advantage of the slot bonding approach in terms of PDR, as it can bond multiple regular slots together in a bonded slot without over-allocating the slot time it needs and thus avoid wasting radio on time and energy. It was also shown that simultaneously using different modulations in the same network compared to only allowing the most reliable modulation demands less radio on time while maintaining the same PDR. Finally, the simulations showed insight in the preferred modulation and parent selections for the slot bonding approach to optimize the number of delivered packets.

In future work, the GA will not be used in real TSCH networks because of the complexity of solving the slot bonding problem, but the lessons learnt of the allocation analysis in this work will be bundled in a heuristic that can be integrated in existing distributed scheduling and routing approaches to allow for fast, but efficient modulation and parent selection choices in multi-MCS TSCH networks. We will also implement these approaches together with the slot bonding technique on real hardware to demonstrate multiple modulations in a single TSCH network. Additional future steps could be the improvement of the current slot bonding formulation by not demanding the constraint of interference avoiding and allowing multiple parallel transmissions as long as no packet loss is expected.

ACKNOWLEDGMENT

Part of this research was funded by the Flemish FWO SBO S004017N Intelligent DEnse And Longe range IoT networks (IDEAL-IoT) project, and by the Internet-of-Shipping (IoS) project. IoS is co-financed by imec and received support from Flanders Innovation & Entrepreneurship. Robbe Elsas is funded by the Flemish FWO SB under grant number 67684. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by FWO and the Flemish Government - department EWI.

REFERENCES

[1] M. Raza, N. Aslam, H. Le-Minh, S. Hussain, Y. Cao, and N. M. Khan, “A Critical Analysis of Research Potential, Challenges, and Future Directives in Industrial Wireless Sensor Networks,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 39–95, 2017.

[2] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Std. 802.15.4-2015, 2016.

[3] *IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer*, IEEE Std. 802.15.4e-2012, 2012.

[4] P. Thubert, “An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4,” Internet Engineering Task Force, Internet-Draft draft-ietf-6tisch-architecture-28, Oct. 2019, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-architecture-28

[5] *IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks*, IEEE Std. 802.15.4g-2012, 2012.

[6] P. Tuset-Peiró, R. D. Gomes, P. Thubert, and X. Vilajosana, “Evaluating IEEE 802.15.4g SUN for Dependable Low-Power Wireless Communications In Industrial Scenarios,” *Sensors (available on Preprints)*, vol. 20, 2020.

[7] G. Daneels, C. Delgado, S. Latré, and J. Famaey, “Towards Slot Bonding for Adaptive MCS in IEEE 802.15.4e TSCH Networks,” in *Proceedings of the IEEE International Conference on Communications*. IEEE, 2020.

[8] J. Muñoz, T. Chang, X. Vilajosana, and T. Watteyne, “Evaluation of IEEE802.15.4g for Environmental Observations,” *Sensors*, vol. 18, no. 10, p. 3468, 2018.

[9] C.-S. Sum, M.-T. Zhou, F. Kojima, and H. Harada, “Experimental Performance Evaluation of Multihop IEEE 802.15. 4/4g/4e smart utility networks in outdoor environment,” *Wireless Communications and Mobile Computing*, vol. 2017, 2017.

[10] J. Muñoz, E. Riou, X. Vilajosana, P. Muhlethaler, and T. Watteyne, “Overview of IEEE802.15.4g OFDM and its Applicability to Smart Building Applications,” in *2018 Wireless Days (WD)*. IEEE, 2018, pp. 123–130.

[11] P. Tuset-Peiró, F. Vázquez-Gallego, J. Muñoz, T. Watteyne, J. Alonso-Zarate, and X. Vilajosana, “Experimental Interference Robustness Evaluation of IEEE 802.15. 4-2015 OQPSK-DSSS and SUN-OFDM Physical Layers,” *arXiv preprint arXiv:1905.11102*, 2019.

[12] R. D. Gomes, P. Tuset-Peiró, and X. Vilajosana, “Improving Link Reliability of IEEE 802.15.4g SUN Networks with Adaptive Modulation Diversity,” in *31st IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2020)*. IEEE, 2020.

[13] P. Tuset-Peiró, F. Adelantado, X. Vilajosana, and R. D. Gomes, “Reliability through Modulation Diversity: Can Combining Multiple IEEE 802.15.4-2015 SUN Modulations Improve PDR?” in *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020, pp. 1–6.

[14] T. Watteyne, S. Lanzisera, A. Mehta, and K. S. Pister, “Mitigating Multipath Fading through Channel Hopping in Wireless Sensor Networks,” in *2010 IEEE International Conference on Communications*. IEEE, 2010, pp. 1–5.

[15] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, “On Optimal Scheduling in Duty-cycled Industrial IoT Applications using IEEE802.15.4e TSCH,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3655–3666, 2013.

[16] E. Municio and S. Latré, “Decentralized broadcast-based scheduling for dense multi-hop TSCH networks,” in *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*, 2016, pp. 19–24.

[17] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust Mesh Networks through Autonomously Scheduled TSCH,” in *Proceedings of the 13th ACM conference on embedded networked sensor systems*. ACM, 2015, pp. 337–350.

[18] P. H. Gomes, T. Watteyne, and B. Krishnamachari, “MABO-TSCH: Multihop and Blacklist-based Optimized Time Synchronized Channel Hopping,” *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 7, p. e3223, 2018.

[19] R.-A. Koutsiamanis, G. Papadopoulos, T. L. Jenschke, P. Thubert, and N. Montavont, “Meet the PAREO Functions: Towards Reliable and Available Wireless Networks,” in *IEEE International Conference on Communications (ICC)*, 2020.

[20] P. Thubert and G. Papadopoulos, “Reliable and Available Wireless Problem Statement,” Internet Engineering Task Force, Internet-Draft draft-pthubert-raw-problem-statement-04, Oct. 2019, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-pthubert-raw-problem-statement-04

[21] Z. Cao, P. Zhou, R. Li, S. Huang, and D. Wu, “Multi-agent deep reinforcement learning for joint multi-channel access and task offloading of mobile edge computing in industry 4.0,” *IEEE Internet of Things Journal*, 2020.

[22] M. Brachmann, S. Duquennoy, N. Tsiftes, and T. Voigt, “Ieee 802.15. 4 tsch in sub-ghz: Design considerations and multi-band support,” in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 42–50.

[23] G. Daneels, E. Municio, B. Van de Velde, G. Ergeerts, M. Weyn, S. Latré, and J. Famaey, “Accurate Energy Consumption Modeling of IEEE 802.15.4e TSCH Using Dual-Band OpenMote Hardware,” *Sensors*, vol. 18, no. 2, p. 437, 2018.

[24] Q. Wang, X. Vilajosana, and T. Watteyne, “6TiSCH Operation Sublayer (6top) Protocol (6P),” RFC 8480, Nov. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8480.txt

[25] C. M. Grinstead and J. L. Snell, *Introduction to probability*. American Mathematical Soc., 2012.

[26] S. C. Ergen and P. Varaiya, “Tdma scheduling algorithms for wireless sensor networks,” *Wireless networks*, vol. 16, no. 4, pp. 985–997, 2010.

[27] D. S. Johnson, J. K. Lenstra, and A. R. Kan, “The complexity of the network design problem,” *Networks*, vol. 8, no. 4, pp. 279–285, 1978.

[28] J. H. Holland, “Genetic algorithms,” *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[29] M. Ojo, S. Giordano, G. Portaluri, and D. Adami, “Throughput maximization scheduling algorithm in tsch networks with deadline constraints,” in *2017 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2017, pp. 1–6.

[30] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[31] E. Municio, G. Daneels, M. Vučinić, S. Latré, J. Famaey, Y. Tanaka, K. Brun, K. Muraoka, X. Vilajosana, and T. Watteyne, “Simulating 6TiSCH Networks,” *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 3, p. e3494, 2019.

[32] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, “DEAP: Evolutionary Algorithms Made Easy,” *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[33] B. Bellekens, L. Tian, P. Boer, M. Weyn, and J. Famaey, “Outdoor IEEE 802.11ah Range Characterization using Validated Propagation Models,” in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[34] M. Lacage and T. R. Henderson, “Yet another network simulator,” in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*. ACM, 2006, p. 12.

[35] X. Vilajosana, K. Pister, and T. Watteyne, “Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration,” RFC 8180, May 2017. [Online]. Available: https://rfc-editor.org/rfc/rfc8180.txt