On *Optimal* Scheduling in Duty-Cycled Industrial IoT Applications using IEEE802.15.4e TSCH

Maria Rita Palattella*, Nicola Accettura†, Luigi Alfredo Grieco†, Gennaro Boggia†,
Mischa Dohler‡ and Thomas Engel*

*SnT, University of Luxembourg (Luxembourg), {maria-rita.palattella, thomas.engel}@uni.lu †DEI, Politecnico di Bari (Italy), {n.accettura,a.grieco,g.boggia}@poliba.it ‡CTTC (Spain), mischa.dohler@cttc.es

Abstract

As exposed in a recent report by General Electric, an Industrial Internet of Things (IoT) is emerging as a commercially viable embodiment of the IoT where physical sensors gather data readings from the field and deliver the traffic to the Internet. The collected real-time "big data", in turn, allows optimizing entire industry verticals with enormous return of investments. Whilst opportunities are ample, it comes along with serious engineering design challenges since industrial applications have stringent requirements on delay, lifetime and standards-compliance. To this end, we advocate the use of an IEEE/IETF standardized IoT architecture along with a recently introduced data-centric scheduling algorithm referred to as Traffic Aware Scheduling Algorithm (TASA). Applying graph theoretical tools to the multi-channel, time-synchronized and duty-cycled nature of TASA, we rigorously derive optimality and bounds on the minimum number of needed active slots (impacting end-to-end delays) and the network duty-cycle (impacting lifetime). We demonstrate the enormous superiority of TASA over traditional IEEE802.15.4/ZigBee approaches in terms of energy efficiency. The outcome of this work is currently laying the foundations of a recently formed IETF standardization group 6TSCH with the aim to significantly improve IoT data flows over IEEE802.15.4e TSCH and IETF 6LoWPAN/ROLL enabled technologies.

Index Terms

IoT Architecture, IoT Standardization, WSN, Sensor Communication Protocol, IEEE802.15.4e, Energy Efficiency, Time Synchronized Channel Hopping, Duty Cycle.

I. Introduction

The proliferation of a huge amount of devices able to be directly connected to the Internet is leading to a new ubiquitous-computing paradigm. This new paradigm enhances the traditional Internet into an Internet of Things (IoT) created around intelligent interconnections of several *smart* objects (sensors, RFIDs, machines, vehicles, smart phones, tablets, etc.) in the physical world. The IoT approach is gaining ground in several application areas, such as smart cities, home automation, industrial applications, among others. Its impact onto economies and societies

around the world is becoming enormous, as highlighted in a recently published report by General Electric showing that operational improvements due to real-time data collected from an Industrial IoT yields Billions of Dollars in return of investment [1].

Flagship applications of such an industrial IoT are telemetry readings of status of oil/brakes/etc of cars on the move; health state measurements of blood pressure/heartbeat of elderly; monitoring of corrosion state of oil/gas pipelines; occupancy measurements of parking in cities; remote metering of water consumption; real-time monitoring of critical parts of a machinery; etc.

Whilst applications are ample, the engineering efforts to enable these innovative services are tremendous. Undisputed, however, is the fact that only a rigorously standardized approach at data transmission, data handling, servicing and policy level will facilitate a viable technology up-take. To this end, in 2003, the IEEE started to work on a framework to federate the communication protocol of the emerging low power wireless sensors and actuator systems. The IEEE802.15.4 protocol [2], which defines low-power Physical (PHY) and Medium Access Control (MAC) layers, is for sure the standard with the longest-standing impact. It has been the foundation of ZigBee 1.0 and ZigBee 2006, the market uptake which however has been well below prior IoT growth predictions; today, this is attributed to its poor technical characteristics effectively preventing true "Internet" capabilities.

Latest standardization efforts aim to circumvent these weaknesses where the IEEE defined a new MAC, the IEEE802.15.4e, based on the Timeslotted Channel Hopping (TSCH) protocol and being similar to the well-proven WirelessHART standard. At the networking layer, the development of low power wireless communication triggered the birth of various IETF working groups: 6LoWPAN to define a convergence layer [3], ROLL RPL [4] to define a routing protocol and CORE CoAP [5] to facilitating native support of current Internet applications over UDP for datagram oriented communication [6]. And, as of February 2013, the new working group IETF 6TSCH aims at bonding IEEE MAC and IETF networking approaches to yield a robust industrial IoT protocol stack [7], [8].

The focus of this work, however, is mainly on the MAC layer where − due to the contentious and delay-critical nature of the Industrial IoT − most energy savings can be achieved. The IEEE802.15.4 low-power radios have been used by the most prominent wireless technology to-date. The limitation of its MAC layer in mesh-networking condition, mainly due to (i) the use of a single channel, and (ii) the high energy waste resulting from the permanently active relay/router nodes, has became apparent only recently. To overcome this limitation, the IEEE802.15e Working Group has defined an amendment of the IEEE802.15.4-2011 standard [2] better suitable for multi-hop networks and able to fulfill industrial communication requirements.

The truly innovative element of the IEEE802.15.4e standard is the TSCH protocol that represents the latest generation of highly reliable and low-power MAC protocols [9], conceived for multi-hop Wireless Sensor Networks (WSNs) based on the Highway Addressable Remote Transducer (HART) technology [10]. Through time synchronization and channel hopping, TSCH enables high reliability while maintaining very low duty cycles and thus utmost power efficiency. It is also well suited for improved multi-hop operations with duty-cycled relay nodes and natural support of overlaying routing protocols, such as the IETF ROLL RPL protocol [4] and the to-be-developed IETF 6TSCH protocols.

With the IEEE802.15.4e standard [11] now finally published, the real challenge is to have protocols and mechanisms in place which allow to build schedules depending on the topology and data traffic requirements. We have been facing this issue, and in [12] we have proposed a novel Traffic Aware Scheduling Algorithm (TASA) which allows building schedules over an IEEE802.15.4e network based on the network topology and the traffic load offered by each node. Industrial applications have stringent requirements on delay, and lifetime. Therefore, TASA while scheduling the transmissions, minimizes the number of active time slots; such number direct impacts the end-to-end delay, and also the network duty cycle, and, thus, the network lifetime. In [13] we have derived some fundamental bounds on the minimum number of active slots for a given topology.

With respect to this prior art, the core contributions of this paper can be summarized as follows:

1) We provide an overview of the two industrial standards, WirelessHART, and ISA100.11a that have paved the way to the TSCH protocol, introducing the concept of timeslotted and channel hopping. In detail, we compare their features, underling differences and analogies with the IEEE802.15.4e standard.
2) We extend the analysis in [13], including all the proofs of Lemmas and Theorem used for computing the bounds on the minimum number of active slots for a given topology, achievable with TASA.
3) We present a preliminary analysis on a possible integration of TASA in a real industrial IoT protocol stack, and we quantify the overhead due to the signaling, i.e., information to be exchanged between the nodes and the master node for setting the schedule up.
4) We show the effectiveness and superiority of the proposed algorithm by means of extensive simulations, where we quantify the network duty-cycle, the average per-node current consumption, as a function of the network size, and the number of child nodes of the master node.

The rest of the paper is organized as follows. In Section II, we compare WirelessHART with ISA100.11a, underlying differences and analogies with the IEEE802.15.4e TSCH. In Section III-A, we describe in detail the novel TASA algorithm, in order to aid the understanding of subsequent sections. In Section IV we provide a proof of Theorem and Lemmas used for computing a lower and an upper bound of the minimum number of active slots necessary for collecting at the *master* node all the data frame generated within a single slotframe. Thus, we compute the duty cycle achievable with TASA. We then conduct in Section V a rigorous performance analysis of TASA, and also a comparison between the traditional IEEE802.15.4 and IEEE802.15.4e. Finally, Section VI concludes the paper.

II. WirelessHART & ISA100.11a: TSCH ancestors

Timeslotted Channel Hopping (TSCH), part of IEEE802.15.4e standard [11], is based on a very simple concept: *synchronize* the nodes for obtaining energy efficiency since minimizing contention, and apply *channel hopping* for increasing reliability since gaining in frequency diversity. Conceptually, this is not a new concept as it was already used in industrial networking technologies such as WirelessHART [10], and ISA100.11a [14].

WirelessHART is the wireless extension of HART, a long standing protocol suite for networking industrial equipment. It is based on the HART Communication protocol, the application layer of which has been in existence since the late 80s. In WirelessHART, communications are precisely scheduled based on Time Division Multiple Access (TDMA) and employ a channel hopping scheme for achieving added data bandwidth and robustness. In detail, WirelessHART uses a $10\ ms$ time slot, and a single algorithm for channel hopping is defined. Most of the traffic in the WirelessHART mesh networks is directed along graph routes. Graphs are a routing structure that creates a connection between network devices over one or more hops and one or more paths. Scheduling is performed by a *centralized network manager* that uses overall network routing information in combination with communication requirements provided by devices and applications. The schedule is translated into transmit and receive slots and transferred from the network manager to each device. Finally, the network manager continuously adapts the network graphs and network schedules to changes in the network topology and communication demand [15].

ISA100.11a was developed through the International Society of Automation (ISA), a U.S.-based and non-profit organization. ISA100.11a is intended to be part of a family of standards designed to support a wide range of wireless industrial plant needs, including process automation, factory automation and RFID. Among the design criteria of ISA100.11a are: flexibility, support for multiple protocols, and multiple applications, use of open standards, reliability (error detection, channel hopping), determinism (TDMA, QOS support), and security [15]. ISA100.11a defines the protocol stack to be used over low-power, low-rate wireless networks, but it does not specify a process automation protocol application layer. The network and transport layers are based on 6LoWPAN [3], IPv6 and UDP standards.

The ISA100.11a data link layer (DLL) is unique to ISA100.11a and uses a non-compliant form of the IEEE802.15.4 MAC. Like the WirelessHART DDL, the ISA100.11a DDL does implement frequency hopping and time slotted time domain multiple access features. But, it is much more flexible in its specifications: time slot size is configurable – $10\ ms$ is only one of the supported slot size – and three channel hopping methods (i.e., slotted, slow, and hybrid) with five channel hopping patterns, are supported [16].

Unlike the WirelessHART DLL, the ISA100.11a data link layer creates and uses graph routing that provides for a number of different data paths for different types of network traffic within the data link subnet. Multiple graphs are used by different devices to transmit different types of data (e.g., periodic data, event data, large blocks of data, etc.). The graphs are built by a *System Manager*, according to the data throughput and transmissions requirements, defined by the system/network designer, and the information provided by the sensor nodes about the RF environment.

IEEE802.15.4e TSCH builds on the same foundations as WirelessHART. But, unlike the industrial protocol which is application-specific, and provides full descriptions of the entire communication stack, TSCH defines the MAC layer only.

In a TSCH network, all nodes are synchronized using a slotframe structure, i.e., a group of slots of equal length which repeats indefinitely over time. The duration of a slot is implementation-specific, like in ISA100.11a, even though a $10\ ms$-length is suggested in the standard [11]. A single slot is long enough for the transmitter to send a maximum-length packet, and for the receiver to send back an acknowledgment indicating the state of reception. Once synchronized, nodes follow a schedule that specifies for each of them which slot and channel offset they can

use for exchanging their data within the slotframe.

Since nodes communicate only when scheduled, they can keep their radios off for most of the time, and thus save energy. Moreover, since more nodes can transmit at the same time, using different channel offsets, it is possible to increase the network capacity. Finally, the frequency diversity implied by channel hopping mitigates the effects of external interference and multipath fading [17]–[19].

The IEEE802.15.4e standard [11] defines how the MAC layer executes a schedule, but it does not specify how such schedule is built. Unlike its ancestors [10], [14], the IEEE802.15.4e TSCH does not impose the use of a centralized solution. Thus, both centralized and distributed approaches can be used for building a schedule that fulfills the specific requirements of the network (e.g., throughput, delay, etc.).

In a centralized approach, a *manager* node (e.g., the gateway node connecting the network to the Internet, playing the role of the WirelessHART *Network Manager*, is responsible for building and maintaining the schedule. In 2009, Fiore et al. have proposed a multi-hop multi-channel scheduling algorithm for wireless control in WirelessHART networks [20]. The suggested heuristic algorithm allocates a link in the first unscheduled time slot, on the first available channel. By doing so, the network requirements may not be satisfied. Therefore, such algorithm, does not seem to be promising for IEEE802.15.4e TSCH networks.

In a distributed approach, nodes decide locally on which links (i.e., slot/channel offset) to schedule with which neighbors. It is more suitable for mobile networks, or networks with many gateway nodes. The simplest solution is for each node to schedule a link to each neighbor. This is the approach adopted by Tinka et al. in [21], and evaluated both by simulation and experimentally.

Whilst in both cases (i.e., centralized and distributed) the schedule needs to be built carefully to ensure a proper network behavior, centralized schedules are known to be superior to distributed ones for fairly static network [22]. Particularly, under increasing load conditions, they offer better performance as it has been corroborated by commercial deployment of thousands of nodes using TSMP [9], [23].

For this reason, we have proposed TASA in [12], a *centralized* scheduling algorithm, for *static*[1] multi-hop IEEE802.15.4e networks. In order to maximize parallel transmissions, TASA allocates time slots and channel-offsets based on the network topology. This approach was also used by Ni and Chen in [24] for their *distributed* MAC scheduling algorithm for Intelligent Transportation Sensor Network. Unlike [24], TASA builds the schedule based also on the data-traffic load offered by each source node to the network. In this way, it allows to obtain better performance in terms of throughput and latency, using a smaller number of channels [12], [24].

III. TASA FOR IEEE802.15.4E TSCH

In this section we introduce the centralized Traffic Aware Scheduling Algorithm (TASA), that we have proposed in [12]. With respect to this prior work,

[1]We do not consider *supporting mobility* because it is not the focus of the IEEE802.15.4e standard, neither of the majority of embedded systems.

- we first detail how TASA could be integrated in an IoT protocol stack, using IEEE802.15.4e TSCH MAC and RPL;
- we then provide all the details of the *Matching* and *Coloring* functions, that were omitted in [12], and we show the different steps of the two TASA procedures, by means of flow charts.

The TASA algorithm is able to fulfill the main requirements in term of delay, duty cycle, and power consumption, of industrial sensor IoT applications. Therefore, it is a potential scheduling candidate for industrial IoT protocol stacks [6], [8], adopting the IEEE802.15.4e TSCH at the MAC layer and the ROLL routing protocol RPL at the network layer. Within the IETF 6TSCH WG, we are currently discussing how TASA can be integrated in a real IoT stack. We have proposed a couple of possible solutions, briefly summarized hereafter. They will be further investigated in future work.

In the emerging 6TSCH architecture [8], TASA could work at the application layer in conjunction with the Path Computation Element (PCE) Communication Protocol (PCEP) [25], a protocol designed specifically for communications between a Path Computation Client (PCC) and a PCE. According to the specifications in [25], the PCE is able to compute a network path or route based on a network graph and applying computational constraints. The path computation algorithmic used by the PCE to optimize a path with respect to a specific metric is outside the scope of the RFC5440. Therefore, TASA could fill this gap, and provide to the PCE (i.e., TASA master node), the algorithm for computing paths (i.e., building the schedule), according to the network topology and the bandwidth (i.e., number of time slots) requested by each node (i.e., PCC).

Note that a PCEP session can be established only over a TCP connection. Therefore, a different approach may be needed for very low-power industrial applications, using UDP at the transport layer [6]. In this case, the TASA algorithm could be adopted in a network management over the Constrained Application Protocol (CoAP) [5]. Once clearly specified, this solution will be integrated, and, thus, tested in the open-source implementation of the IoT protocol stack, developed by the OpenWSN project [26].

Regardless the application protocol used for sending the schedule from the master node to the nodes, the 6TUS adaptation protocol [27], developed within the 6TSCH WG, will provide the interface for the PCE to install the schedule built with TASA, into the network, using specific commands and messages. Moreover, the 6TUS adaptation protocol provides also all the needed functionalities for setting up the network, task that is out of the scope of TASA.

A. Network Model and Notation

The Traffic Aware Scheduling Algorithm (TASA) has been designed for building efficient schedules for data gathering networks, such us wireless sensor networks (WSNs) [28]. Thus, it is applied to a network with a tree topology that can be represented using an oriented graph $G = (V, E)$, where $V = \{n_0, n_1, \ldots, n_{N-1}\}$ is the set of devices, and $|V| = N$ is the total number of nodes in the network. In particular, n_0 is the PAN coordinator, acting both as *master node* and collecting node; and n_i, with $1 \leq i \leq N-1$, is the generic i-th node of the network (FFD or RFD device [2]). Table I summarizes the notation used for describing the TASA algorithm [12].

TABLE I
TASA ALGORITHM NOTATION

Symbol	Description
N	total number of nodes in the network
n_i	i-th node in the network
p_i	parent node of n_i
z_i	number of neighbors of n_i
h_i	hop distance between n_i and the *master* node
$ch(n_i)$	set of child nodes of n_i
$ST(n_i)$	sub-tree having n_i as root node
DC_i	set of links in *duplex-conflict* with (n_i,p_i)
$DCFL(k)$	set of *duplex-conflict-free* links, selected for slot k
$ICFL_c(k)$	set of *interference-conflict-free* links, assigned to channel offset c in slot k
S	slotframe size
λ	minimum number of slots necessary for scheduling all the transmissions
λ	actual number of active slots necessary for scheduling all the transmissions (returned by simulation)
G	Graph
V	set of vertices $\in G$ and P, i.e., nodes in the network
E	set of edges $\in G$, i.e., *dedicated links* in the network
P	Physical Connectivity Graph
C	set of edges between connected nodes in P
$I(k)$	*Interference Conflict* Graph built for slot k
$V_I(k)$	set of vertices $\in I(k)$
$E_I(k)$	set of *interference-conflict* edges $\in I(k)$
$\tilde{q}_i$	integer number of packets generated by node n_i within a slotframe
$\tilde{Q}$	total number of packets generated within a slotframe by all the nodes in the network
$q_i(k)$	local queue level of node n_i in slot k
$Q_i(k)$	global queue level of $ST(n_i)$ in slot k

For each node n_i, we define: (a) its parent node, p_i; (b) the set of its child nodes, $ch(n_i)$[2]; (c) its sub-tree $ST(n_i)$, composed by n_i itself and all the nodes connected to it through multi-hop paths. All the nodes are synchronized with the same slotframe and they follow a common schedule. In order to provide a reliable schedule, and thus avoid collision, we assume that each node n_i is connected to its parent p_i with a dedicated link $(n_i, p_i) \in E \subset V \times V$, [6], [11]. Therefore, we have $|E| = N - 1$. It has to be noticed that even though every node n_i is connected only with its parent node p_i, it could have more neighbors and it could switch among them if there is the need. In fact, TASA has been designed, assuming that the gradient-based IETF Routing Protocol for Low-power and lossy networks (RPL) [6] operates at the network layer, on top of the IEEE802.15.4e MAC. RPL organizes the considered physical topology as a Destination Oriented Directed Acyclic Graph (DODAG) having its unique root located at the PAN coordinator. Note that a DODAG is not a tree, because each node in the network maintains a set of possible DODAG parents. In a TASA-based TSCH network a unique preferred parent will be chosen among the different parents, as next-hop for packet transmissions towards the root, by means of an objective function. In detail, the preferred parent will be the one that allows to optimize the routing path according to some objectives, such as minimum latency, maximum throughput, minimum energy consumption. In the TASA network scenarios

[2]Hereafter, we refer to all the nodes belonging to $ch(n_i)$ as *brother* set.

we assume that the *Minimum Rank with Hysteresis Objective Function* [29] is used. The latter updates the next-hop entry in the routing table of the node, only if the achievable routing improvement is greater than a given threshold. By setting the threshold in a appropriate way, it is possible to keep the routing paths stable. Thus, the routing topology becomes almost a tree rooted at the PAN coordinator.

TASA is a traffic-aware algorithm that builds the schedule based on the traffic offered by every node to the network. In detail, we assume the network supports a *multi-point-to-point* traffic: each node n_i, with $i \neq 0$, generates a constant[3] integer number of packets, $\tilde{q}_i$, within a slotframe with a size equal to S slots, destined to the root node, n_0. Thus, each source node n_i forwards its data frame to its parent node p_i in the tree rooted at the PAN coordinator.

Let (a) $\tilde{Q}$ be the network traffic load, i.e., the total number of packets delivered to the PAN coordinator within a single slotframe, which results in $\tilde{Q} = \sum_{i=1}^{N-1} \tilde{q}_i$; (b) $q_i(k)$ be the node *local queue level* in a given slot k, i.e., the number of packets that are in the queue of the node n_i in the slot k (e.g., $q_i(k) = \tilde{q}_i$ if $k = 0$); and (c) $Q_i(k)$ be the *global queue level* of a node in slot k, i.e., the total number of packets that are in the queues of the nodes belonging to the given sub-tree in that slot k [12].

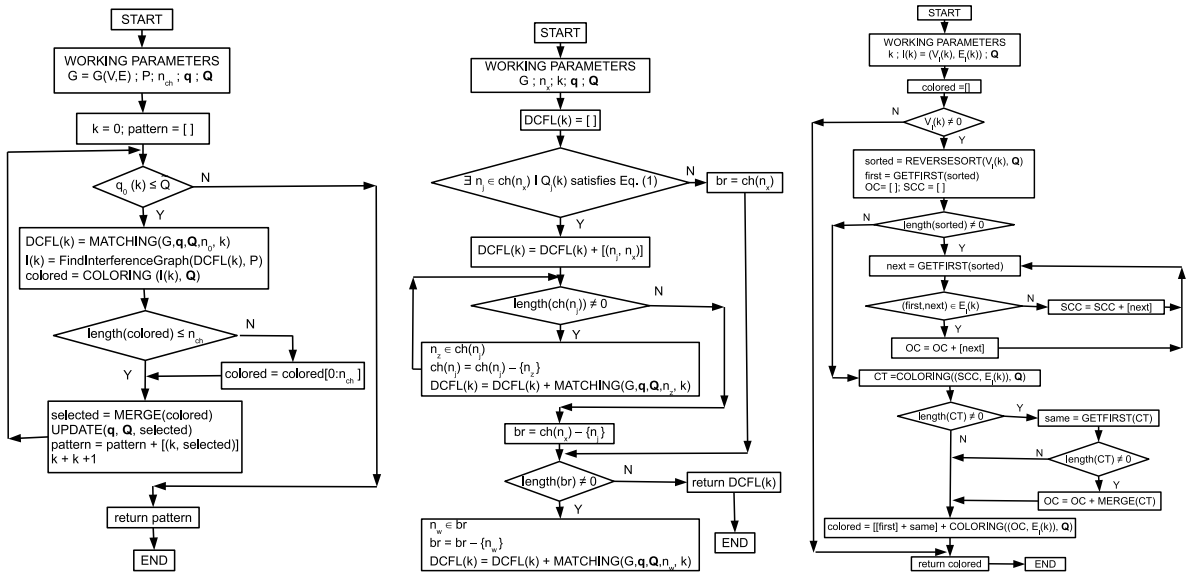
TASA builds the schedule, making sure that no conflicts, either duplex or interference, will happen in the network. We define the set of *duplex-conflict* links, DC_i, for a given node n_i, the sets of edges between node pairs in G that cannot transmit in the same slot reserved to the link (n_i, p_i). Obviously, only sets of *duplex-conflict free* links, i.e., $DCFL(k)$, can be scheduled in the same slot k within the slotframe. Similarly, we define as *interference-conflict* links those that interfere if they are scheduled at the same time on the same channel offset. Let $ICFL_c(k)$ be the set of *interference-conflict free* links that can use the same generic channel offset c in a given slot k.

It has to be noticed that, beyond the mathematical formulation of the scheduling problem, both sets of *duplex-conflict-free* links and *interference-conflict-free* links can be obtained in practice, using the *Matching* and *Coloring* procedure, described in the following Section. To this aim, the master node needs to have complete knowledge of the network topology and the bandwidth (i.e., number of time slots) requested by each source node. Even though the way in which the needed information is gathered is out of the scope of this work, we can imagine that for each node n_i, with $i \neq 0$, the master node collects information about (i) the physical neighbors, (ii) the selected parent, and (iii) the traffic generated within a single slotframe. The routing protocol RPL could provide support for gathering the information related to the network topology, while the application protocol, e.g., PCEP, could help in specifying the path requirements (i.e., requested bandwidth), using the PCReq messages [25]. These aspects will be further investigated and better defined in our future work.

B. TASA main procedures: Matching and Coloring

The flow chart in Fig. 1(a) summarizes the main steps of TASA, running on the *master* node. The latter has complete topology information, i.e., it knows the graph G and the physical connectivity graph P (i.e., physical

[3]Our assumption is based on the fact that traffic is typically different between nodes but relatively constant over time in emerging heterogeneous embedded applications.



(a) TASA algorithm (b) *Matching* procedure (c) *Coloring* procedure

Fig. 1. Flow chart of the main procedures of the TASA algorithm.

neighbors and selected parent, p_i, $\forall\ n_i \in V$, with $i \neq 0$); the number of available channels, n_{ch}; and finally, the traffic load generated by each node ($\tilde{q_i}$, $\forall\ n_i\ \in V$, with $i \neq 0$). Starting from such information, the *master* node can build the schedule, implementing in an iterative way the *matching* and *coloring* procedures[4] (shown respectively in Fig. 1(b) and Fig. 1(c)).

In detail, *Matching*, described in Fig. 1(b), is used for selecting a $DCFL(k)$ set for a given slot k. Note that a set of *duplex-conflict-free* links includes several subsets of *interference-conflict-free* links, and each of them has to be scheduled on a different channel offset. In order to select these $ICFL_c(k)$ sets, with $1 \leq c \leq n_{ch}$, an *Interference Conflict* Graph, $I(k) = \{V_I(k), E_I(k)\}$ is built, where $V_I(k) \subset V$ is the set of transmitting nodes n_i of the dedicated links (n_i, p_i) belonging to $DCFL(k)$, and $E_I(k) \subset C$ is the set of interfering links. Applying the *coloring* procedure in Fig. 1(c), to $I(k)$ the *master* node can select the $ICFL_c(k)$ subsets belonging to $DCFL(k)$, that have to be scheduled on the same channel offset c. Note that n_{ch} could be smaller than the number of $ICFL_c(k)$ subsets. In this case only some of the links in $DCFL(k)$ will be scheduled in k, while the other ones will be re-considered in the next step of the procedure.

At the end of each step, based on the schedule built for the considered slot k, local and global queue levels are updated [12]. The update assures that the links to be scheduled in the next slot $k+1$, will be chosen according to the traffic load that each device has still to deliver to its parent node. Obviously, the TASA algorithm ends when a schedule that allows to deliver all the network traffic to the root node (i.e., $q_0(k) = \tilde{Q}$) has been built.

As shown in Fig. 1(b), when the matching procedure is applied to a node n_x, it selects among its children,

[4]In fact, the time/channel-offset scheduling problem can be formulated as the combination of *matching* and *vertex coloring* problems in graph theory [12], [30].

$ch(n_x)$, the child node n_j for which it results:

$$Q_j(k) = \max\{Q_i(k) \mid n_i \in ch(n_x) \wedge \ q_i(k) \neq 0\}. \quad (1)$$

Eq. (1) states that a slot k is reserved to a link (n_j, n_x) only when the child node n_j has at least a packet to transmit to its parent n_x [5] during that slot k. In this way, we avoid to reserve resources for nodes whose queues are empty. Thus, we reduce the energy consumption because a node is activated only when it has really data to transmit. Moreover, in order to save energy, the $DCFL(k)$ set is chosen according also to the global queue level, $Q_i(k)$. In this way, it is more lucky that consecutive slots are reserved to the same link (n_i, p_i) through which a large amount of traffic, generated by the nodes in the sub-tree $ST(n_i)$, has to be transmitted. Once the link (n_j, n_x) is chosen, all the links that are in duplex-conflict with it, i.e., that belong to DC_j, cannot be scheduled in the same slot. Therefore, the *matching* procedure is applied to all the conflicting nodes, i.e., the nodes in the brother set (i.e., *br*). The *matching* function, applied in recursive way, after crossing the whole tree, returns the $DCFL(k)$ set, selected for that slot k. Once the $DCFL(k)$ set has been chosen, the corresponding *Interference Conflict* graph $I(k)$ is built and passed as input to the *coloring* procedure. The latter, shown in Fig. 1(c), uses a heuristic incremental method for coloring $I(k)$. First of all, the nodes n_i in $V_I(k)$ are ordered according to $Q_i(k)$ increase reversed order. In order to give priority to more congested links, and thus assure that nodes with more traffic load can transmit in that slot k, the first color c is assigned to the node $n_i \in V_I(k)$ that has the largest *global queue* level, $Q_i(k)$. The remaining nodes are divided in two groups: those that do not interfere with n_i (i.e., *SameColorCandidates*, SCC), and those that interfere with n_i (i.e., *OtherColors*, OC). The next vertex to be colored with the same color c is the first in the SCC list. The remaining nodes in this list are again divided in two groups according to the same criteria, described above. Proceeding in this way, the set of links that can be scheduled on the same channel offset c, in slot k, $ICFL_c(k)$, is built. Finally, applying the same procedure to the links that have been discarded step by step because they interfere with those belonging to the $ICFL_c(k)$ set, (i.e., those in the OC list), it is possible to built the other set of *interference-conflict-free* links to be scheduled on different channels.

Using the schedule built with TASA, which avoids any kind of duplex and interference conflict, and assuming the ideal nature of the radio medium, a TSCH network can achieve a Packet Delivery Ratio (PDR) equal to 100%. Actually, in a real scenario, the inherent variability of the radio medium, combined with possible link failures and network topology changes[6] implies packets losses and thus, compromise the PDR guaranteed by the computed time/frequency schedule. Note that TASA is already tailored for taking into account packet losses. In fact, even though in its actual version, it reserves a number of time slots along a multi-hop path equal to the traffic load offered to the network ($\tilde{q}_i$, $\forall n_i \in V$, with $i \neq 0$), it could allocate more resources (i.e., more time slots per each source node), in order to allow retrasmissions when needed. Finally, note that packets losses could be detected by the *master* node. In fact, being the network manager, it knows the network traffic load that is generated within

[5] $n_x \equiv p_j$, according to the notation adopted in the paper.

[6] For instance, a topology change can be due to the RPL *Objective function* that selects for a given node another DODAG parent as preferred one, and thus updates the next-hop in the multi-hop path toward the master node.

a slotframe, $\tilde{Q}$, and it can monitor the effective number of packets that have been delivered at the end of each slotframe. If for some time (e.g., several slotframe occurences[7]) packets have not been delivered from a portion of the network, the *master node* can conclude that links failures or topology changes happened in the network. Thus, it will collect the new information related to the current network topology and traffic load, and run the TASA algorithm again, in order to build a new schedule.

IV. Fundamental Network Duty-Cycle Bounds

In order to fulfill the strict requirements of emerging industrial IoT applications, the suggested TASA algorithm builds schedules that minimize the duty cycle, thus reducing packets latency and power consumption at the same time. In detail, for delivering the network traffic load, $\tilde{Q}$, to the PAN coordinator, TASA schedules all the transmissions in the first λ slots of the slotframe, with $\lambda \leq S$, leaving the remaining $S - \lambda$ slots empty. Thus, a duty cycle equal to λ/S can be achieved.

In [13] we have provided a lower and upper bound of λ, assuming that the number of available channel offsets, n_{ch}, is enough for coloring the *Interference Conflict* graph $I(k)$ built for each slot k. In this section we significantly extend this analysis, including all the proofs of Lemmas and Theorem used for computing the minimum value of λ, achievable with TASA. The latter is a function of network topology and traffic load generated by each source node, i.e., the two parameters used by TASA for building the schedule.

Lemma 1: Assuming that each node $n_i \in V$, with $i \neq 0$, generates within a single slotframe $\tilde{q}_i$ packets, and $\tilde{Q} = \sum \tilde{q}_i$. The minimum number of active slots λ is equal to $\tilde{Q}$.

Proof: The PAN coordinator can receive at most one packet in each slot. Therefore, at least $\tilde{Q}$ slots are necessary for all packets to reach the root node. This gives us the lower bound of λ. ■

Note that λ depends on the initial value at $k = 0$ of the *global queue* levels. In fact, given that a packet can move inside a sub-tree from node to node only during a single slot, $Q_i(k)$ can be unchanged or decrease in time, while data frames are moved one-hop closer to the root node.

Moreover, λ is mainly affected by the traffic delivered from the child nodes of the PAN coordinator, due to the rule they play (i.e., collect all the packets generated in the network and then forward them to the root node). In detail, due to the duplex-conflict, in each slot only one node in the brother set $ch(n_0)$ can transmit a data frame to the root node. In the same slot, the other nodes in the aforementioned set can only receive packets from their children.

In the optimal operative condition, i.e., when the traffic load is well spread among the different nodes in the network, it is possible to assure that a packet is delivered to the PAN coordinator, while at least another data frame is moved one-hop closer to the root node. This condition is not fulfilled when one of the child nodes of the PAN coordinator, $n_j \in ch(n_0)$, receives from the other nodes in its sub-tree $ST(n_j)$ a number of packets greater than

[7]A time out for trigger the computation of a new TASA schedule can be fixed in a real scenario, based on the requirements of the specific industrial application.

the total number of data frames that the other child nodes $n_z \in ch(n_0)$, with $z \neq j$, will transmit to the PAN coordinator, within one slotframe. In other words, it results:

$$Q_j(0) - \tilde{q_j} > \sum_{\substack{n_z \in ch(n_0) \\ z \neq j}} Q_z(0) \ . \tag{2}$$

Adding $Q_j(0)$ to both sizes of the inequality, Eq. (2) can be equivalently rewritten as follows:

$$2 \cdot Q_j(0) - \tilde{q_j} > \tilde{Q} \ . \tag{3}$$

Before computing the actual value of λ in the aforementioned condition, we introduce the following lemma.

Lemma 2: Assume that each node $n_i \in V$, with $i \neq 0$, generates within a single slotframe $\tilde{q_i}$ packets, and that the PAN coordinator has M child nodes, i.e., $ch(n_0) = \{n_1, n_2, ..., n_M\}$, each of them having an initial global queue level $Q_i(0)$, with $1 \leq i \leq M$. Let $\tilde{Q} = \sum_{i=1}^{N-1} \tilde{q_i} = \sum_{i=1}^{M} Q_i(0)$. Then, there is only one child node that can satisfy Eq. (3).

Proof: Let assume that the statement in *Lemma* 2 is false, and that two children of the root node, i.e., n_1 and n_2, satisfy Eq. (3). In other words, it results

$$2Q_1(0) - \tilde{q_1} > \tilde{Q}$$

$$2Q_2(0) - \tilde{q_2} > \tilde{Q}$$

Then, it follows $2Q_1(0) + 2Q_2(0) - \tilde{q_1} - \tilde{q_2} > 2\tilde{Q}$, or equivalently

$$-(\tilde{q_1} + \tilde{q_2}) > 2\sum_{i=3}^{M} Q_i(0)$$

that is contradicting, because $\tilde{q_i} > 0$, and $Q_i(k) \geq 0$, $\forall\ i$. ■

Theorem 1: Assume that the number of available channel offsets n_{ch} is enough for coloring the *Interference Conflict* graph, $I(k)$, built for every slot k. Each node $n_i \in V$, with $i \neq 0$, generates $\tilde{q_i}$ packets in a single slotframe, and $\tilde{Q} = \sum \tilde{q_i}$. Assume also that there exists a child node of the PAN coordinator, $n_j \in ch(n_0)$, that receives from the other nodes in its sub-tree $ST(n_j)$ a number of packets larger than the total number of data frames that the other child nodes $n_z \in ch(n_0)$, with $z \neq j$, will transmit to the *master* node, within one slotframe. Then, the minimum number of active slots λ is equal to $2 \cdot Q_j(0) - \tilde{q_j}$.

Proof: The PAN coordinator can receive at most one packet in each slot. Thus, the node n_j, fulfilling Eq. (2), needs $\tilde{q_j}$ slots for transmitting its local packets to the root node. The other child nodes of the root node, due to the duplex-conflict, cannot transmit at the same time with n_j. Therefore, other $\sum Q_z(0)$ slots are necessary for allowing the root node to receive the data frames generated in the sub-trees $ST(n_z)$, with $n_z \in ch(n_0)$ and $z \neq j$. Similarly, during the aforementioned slots, the node n_j cannot transmit, but it can receive the same amount of data from its children. In order to transmit these received packets it needs others $\sum Q_z(0)$ slots. Based on the amount of traffic generated by each node in the network, and thus on Eq. (2), the node n_j still have other $Q_j(0) - \tilde{q_j} - \sum_{z \neq j} Q_z(0)$

packets to receive and then to transmit to the root node. Therefore, in the end, the total number of slots necessary for delivering all the packets to the PAN coordinator is given by:

$$\lambda = \tilde{q_j} + 2 \cdot \sum_{z \neq j} Q_z(0) + 2 \cdot [Q_j(0) - \tilde{q_j} - \sum_{z \neq j} Q_z(0)] =$$
$$= 2 \cdot Q_j(0) - \tilde{q_j}. \quad (4)$$

■

Based on the aforementioned Lemmas and Theorem, we can finally conclude that using TASA, the number of active slots λ within a single slotframe, is given by

$$\lambda = \begin{cases} 2Q_j(0) - \tilde{q_j} & \text{if } \exists n_j \in ch(n_0) \text{ fulfilling Eq. (2)} \\ \tilde{Q} & \text{otherwise.} \end{cases} \quad (5)$$

This fundamental insight, functionally relating the number of nodes, topology and data traffic to the system's duty-cycle, allows a proper dimensioning of industrial sensor IoT applications with stringent constraints in delay and power consumption.

V. Performance Evaluation

In order to evaluate the performance that a IEEE802.15.4e TSCH network can achieve with our TASA algorithm, we have developed an ad-hoc simulator in Python language. In detail, according to the TASA network model, we have simulated a TSCH network with a tree topology, deployed in an area of $200 \times 200\text{m}^2$. Each node, located in a random position in the network, has a coverage range $R = 50$m, and it can have a number of neighbors that varies in the range [2, 20]. Because TASA builds the scheduling based on the network topology, we analyze the performance in several scenarios, varying the total number of nodes, $N \in [20, 80]$. Moreover, given the key role that the child nodes of the master node, i.e., $ch(n_0)$, play in the algorithm – according to Eq. (5) – we consider different topologies with $|ch(n_0)| \in [2, 10]$.

TASA allocates time slots/channel offsets based not only on the network topology, but also on the traffic load. For this reason, we test the algorithm under different traffic conditions. In detail, we assume that the average number of packets, $\bar{n}_{pkt}$, generated by each node within a single slotframe can vary in the range [3, 5] [8].

Note that, in this work, by considering different values of $|ch(n_0)|$, and $\bar{n}_{pkt}$, we extend the preliminary performance analysis of TASA, provided in [12], [13].

In our simulations we have fixed $S = 720$ slots to assure that the slotframe is long enough for delivering to the master node the maximum traffic load, offered to a network with maximum size $N = 80$. Moreover, as suggested in [11], we have assumed a time slot-length equal to $10\ ms$. Such duration allows the transmission of a 127-byte PDU. Finally, from the channel hopping point of view, we assume a number of available channels $n_{ch} \in [2, 16]$.

[8] In other words, at the beginning of the slotframe each node generates a number of packets, randomly selected respectively in the range [1, 5], [1, 7] and [1, 9].

A. Verification of Optimality

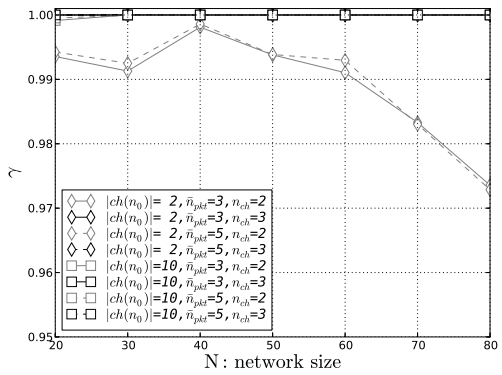


Fig. 2. Match $\gamma = \lambda/\bar{\lambda}$ of theoretically achievable duty cycle $\bar{\lambda}$ and achieved TASA duty cycle λ, where $|ch(n_0)|$ is the number of child nodes of the root.

In order to quantify TASA algorithm efficiency, we define the γ metric as the ratio between λ, i.e., the number of active slots within a slotframe, given by Eq. (5), and the actual number of active slots, $\bar{\lambda}$, from the simulated TASA implementation, i.e., $\gamma = \lambda/\bar{\lambda}$. We have computed the γ metric in different network scenarios, for several values of $|ch(n_0)|$, $\bar{n}_{pkt}$, and n_{ch}. As shown in Fig. 2, the average TASA efficiency is $\gamma > 0.97$, with $|ch(n_0)| = n_{ch} = 2$. Instead, for $|ch(n_0)| = 10$ and/or $n_{ch} \geq 3$, we get $\gamma = 1$, thus obtaining the minimum duty cycle (i.e., the optimal performance), achievable with TASA. Note that a larger number of child nodes, $ch(n_0)$, allows to distribute the traffic load among the different subtrees, $ST(ch(n_0))$, thus to parallelize more transmission at the same time and thereby reducing the number of active time slot per slotframe.

Based on our analysis, the performance does not improve with $n_{ch} \geq 3$. Therefore, we will present the results only for $n_{ch} = 2$ and $n_{ch} = 3$. The worthy feature of reaching the maximum theoretical efficiency with only few channels makes TASA very suitable for emerging IoT applications. In fact, it allows coexistence among several networks (i.e., groups of smart objects), each of them using a different subset of frequencies, chosen among the 16 available channels.

B. Network Performance

To fulfill the requirements of several duty-cycled IoT industrial applications, TASA builds optimal scheduling that allows reducing the network duty cycle, while keeping high throughput. In this section, in order to verify TASA efficiency, we show its performance in terms of network duty cycle and maximum achievable throughput.

We define the average network duty cycle as $DC = \bar{\lambda}/S$, where λ is the number of active slots in a slotframe and S is the slotframe size. As we can see in Fig. 3, this performance index does not depend significantly on the number of available channels, but mainly depends on the network traffic load. In detail, DC grows up with the network size, N, because a larger number of active slots λ is needed for delivering more traffic to the root node. For

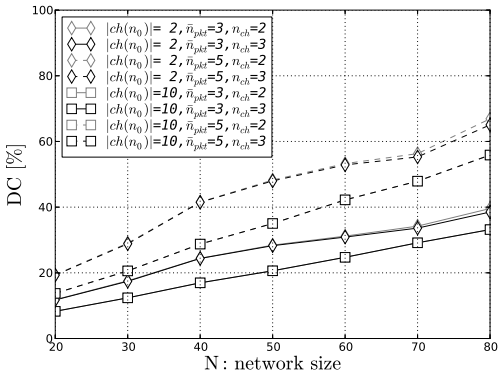


Fig. 3. Network duty cycle for different network sizes.

the same reason, for fixed $|ch(n_0)|$ and n_{ch}, DC increases by almost 15% when each node generates on average 5 instead of 3 packets. In order to reduce the network duty cycle, a larger number of $ch(n_0)$ has to be used. In fact, having more nodes acting as relays, reduces their congestion level thus requiring a shorter time to deliver to the PAN coordinator all the traffic generated at the beginning of the slotframe. In detail, as we can see in Fig. 3, in all the considered operative condition, DC is smaller than 50% in networks with size $N < 50$. The low network duty cycles provided by TASA enable the coexistence of several network instances synchronized with different slotframes, whose active slots work at different time intervals. This is not only facilitated by the IEEE802.15.4e standard but also suits the emerging IoT, allowing several networks to be connected that share the same resources.

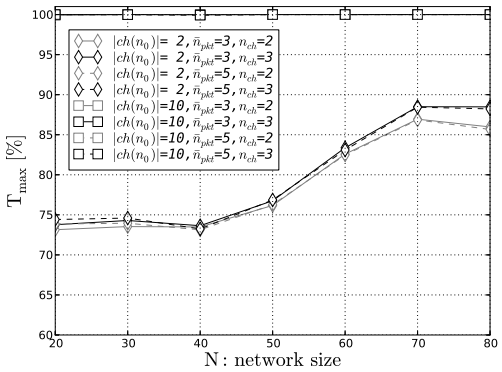


Fig. 4. Maximum achievable throughput, equal to $\tilde{Q}/\bar{\lambda}$.

Finally, we have computed the maximum achievable throughput, T_{max}, defined as $\tilde{Q}/\bar{\lambda}$. As shown in Fig. 4, for $|ch(n_0)| = 2$ the throughput increases with the network size N, i.e., with the number of source nodes generating traffic. Instead, it results $T_{max} = 100\%$ in network topologies having a larger number of child nodes $ch(n_0)$ (i.e., for $|ch(n_0)| = 10$), regardless the number of available channels and the average traffic load. In such scenarios,

the traffic will be spread over the whole network, among several subtrees $ST(ch(n_0))$. It will therefore be very unlikely to have a child node of the master node, satisfying the condition of Theorem 1, thus yielding $\bar{\lambda} = \tilde{Q}$.

C. Signaling overhead

As explained in Sec. III-A, we are still investigating the protocols to be used for (i) gathering at the master node the information needed for setting up the schedule, and (ii) pushing the schedule built by TASA into the network. Even though there is still some work to be done, in this section we present a preliminary analysis of the overhead due to the signaling. In other words, we quantify the amount of traffic (in bytes) that has to be exchanged in the network every time that a new schedule has to be built, e.g., at every link failure or topology change occurrence.

According to the TASA algorithm, the master node needs to know about the network topology and the traffic load generated by each node in the network. Then, once the schedule is built, each node has to be informed about the time/frequency pattern to follow. During the *information gathering* phase, each node $n_i \in V$, with $i \neq 0$, will communicate to the *master* node, n_0, (a) the list of its physical neighbors (i.e., their ID), specifying which one among them is the selected parent, p_i, and (b) the amount of traffic, $\tilde{q}_i$, generated within a slotframe. Let us define z_i the number of physical neighbors of n_i, and let assume that each neighbor is identified with the IEEE802.15.4 2-byte short address [2]. Therefore, the list of neighbors is long $2 \cdot z_i$ bytes. Finally, a byte is needed for indicating the selected parent within the list of neighbors (i.e., an integer $\in [0, z_i - 1]$) and another byte for specify the amount of packets $\tilde{q}_i$ offered to the network.

During the *schedule distribution* phase, each node n_i will receive from the *master* node the list of cells, i.e., time slots and channel offsets, that have been reserved for its transmissions/receptions (TX/RX). We figure out that 2 bytes are sufficient for identifying a cell within a slotframe. In fact, we need: (i) 11 bits for indicating the time slot offset (i.e., the position of the cell from the beginning of the slotframe); (ii) 4 bits for identifying the channel offset chosen among the 16 available ones; (iii) 1 bit/flag for distinguishing between TX and RX.

Note that, according to TASA traffic model described in Sec. III-A, for each node n_i, the *master* node will schedule $2 \cdot Q_i(0) - \tilde{q}_i$ cells, in detail $Q_i(0) - \tilde{q}_i$ in RX and $Q_i(0)$ in TX.

Thus, it follows that the amount of signaling overhead associated to each node n_i is equal to $2 \cdot (z_i + 1 + 2Q_i(0) - \tilde{q}_i)$ bytes. Assuming that the signaling information is forwarded hop-by-hop, along the multi-hop path between the node and the master node, the average per-node signaling overhead, expressed in bytes, is given by:

$$OH = \frac{2}{N} \sum_{i=1}^{N} h_i \cdot (z_i + 1 + 2Q_i(0) - \tilde{q}_i), \tag{6}$$

where h_i is the hop-distance between n_i and n_0.

In Figure 5 we show the average per-node signaling overhead, $\overline{OH}$, in the analyzed network scenarios.

The overhead increases with the average traffic load offered by each node to the network. In fact, the *master* node, will have to push more signaling information into the network for setting the schedule up.

Instead, OH decreases when there is a bigger number of child nodes $ch(n_0)$ in the network. In fact, having a

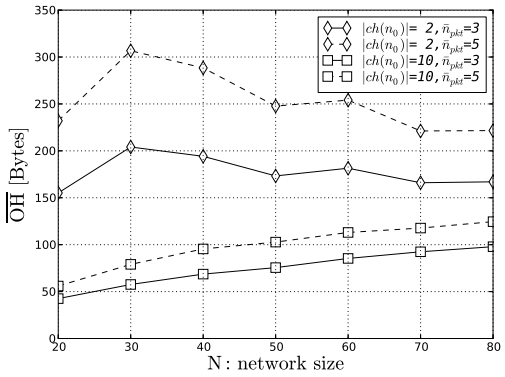


Fig. 5. Average per-node signaling overhead expressed in bytes.

larger $|ch(n_0)|$ allows to reduce the tree graph depth, and therefore, the maximum value of of h_i, with $i = 1, \ldots, N$, in Eq. (6). In other words, the signaling information, will have to travel through a smaller number of hops.

D. TASA-based IEEE802.15.4e vs. IEEE802.15.4

In this section, we compare the performance of networks implementing the IEEE802.15.4e TSCH MAC jointly with our TASA algorithm versus the traditional IEEE802.15.4 MAC protocol. The network performance has been evaluated in terms of loss rate, and average current consumption, that can be easily translated in network lifetime.

In order to analyze the behavior of IEEE802.15.4 networks, we use the Markov chain model proposed by Di Marco et al. [31] for modeling the unslotted CSMA/CA mechanism. It has to be specified that after an in-depth investigation of related prior work, we have chosen the aforementioned model because (a) it is one of the more accurate ones, and (b) mainly it considers a network scenario similar to that analyzed in the present work (i.e., multi-hop tree network, heterogeneous traffic, interaction with routing protocol and ROLL standardization). We have set the IEEE802.15.4 MAC parameters as suggested in [31]. In particular, because we consider only dedicated links in our TASA scheme, we have assumed that no retransmissions are allowed in the IEEE802.15.4 unslotted CSMA/CA mechanism, in order to perform a coherent comparison.

The final aim of the TASA scheme is to set up a collision-free time/frequency schedule. In order to reach this goal, matching and coloring functions play a key role in avoiding simultaneous transmissions on links not belonging to the same $DCFL(k)$ and $ICFL_c(k)$ set. Instead, the classical contention-based IEEE802.15.4 MAC allows collisions and therefore implies a possible loss-rate per hop. Fig. 6 shows the substantial gain of a TASA-based IEEE802.15.4e TSCH network in terms of loss-rate compared to the same network scenario using the IEEE802.15.4 unslotted CSMA/CA, when the source nodes generate on average 3 packets per slotframe.

From Fig. 6, we can notice that while the IEEE802.15.4e TSCH MAC assures no losses (assuming an ideal radio medium), with the IEEE802.15.4 MAC there is a non-negligible end-to-end loss ratio that increases with (i) increasing network size N and (ii) larger number of child nodes $ch(n_0)$, all contending for the same channel for

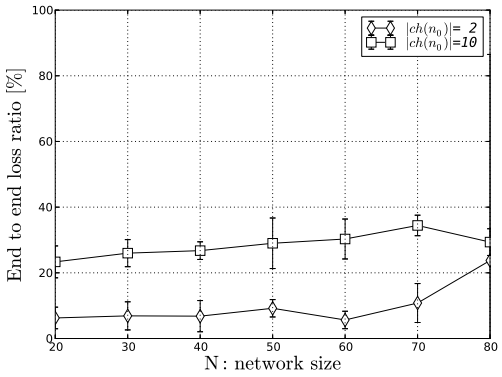


Fig. 6. Loss rate obtained by implementing the IEEE802.15.4 MAC.

transmitting the collected data to the root node n_0. Clearly, from these results, we can conclude that the suggested TASA algorithm, with its loss ratio equal to zero, can be adopted in critical industrial applications that need a high level of reliability.

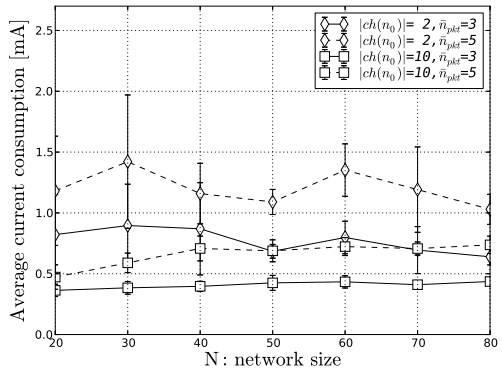


Fig. 7. Average per-node current consumption.

Finally, in order to show the energy-efficiency of TASA, we have computed the average per-node current consumption, that directly impacts the node lifetime. For illustrative purpose, we assume each node is powered by a pair of AA batteries, holding 3000 mAh of charge, and is equipped with a TI CC2430 radio [32], which draws $\approx$ 27 mA when on. Using the IEEE802.15.4 MAC protocol that keeps the radio on all the time (i.e., 100% node duty cycle), the batteries will be depleted in 3000 $mAh/27\ mA = 111\ h$, i.e., almost 5 days. Instead, if on the same hardaware we use TSCH-TASA, which runs the nodes at low duty cycle, the average per-node current consumption will be considerably smaller. In detail, it will be $< 1.5\ mA$ also in the worst operative condition (i.e., $|ch(n_0)| = 2$, and $\bar{n}_{pkt} = 5$), as shown in Fig. 7. Thus, the node lifetime increases to 3000 $mAh/1.5\ mA = 2000\ h$, i.e., $\approx$ 3 months. Replacing the TI CC2430 with another low-power radio, it is possible to further increase the node lifetime.

We can conclude that the use of the TASA algorithm in an IEEE802.15.4e TSCH network implies a gain in term of current consumption, over the same network running the IEEE802.15.4 MAC. The gain is due to the small average per-node duty cycle achievable with TASA and to the TSCH synchronization mechanism. In fact, the lack of such technique in IEEE802.15.4 networks imposes the radio to stay on for all the time, overhearing the wireless medium until a transmission is detected, and thus, wasting additional energy.

VI. Conclusions

The emerging Industrial IoT is based on a standardized communications stack, notably embedded standards from the IEEE and the IETF. Central to these developments is an industrially viable MAC, able to support sensor topologies which scale in size and density; to this end, the IEEE recently ratified the TSCH-based IEEE802.15.4e MAC. Said MAC specifies transceiver operations but no scheduling nor any other network-wide operations.

We had thus introduced a Traffic Aware Scheduling Algorithm (TASA), a centralized scheduling algorithm which exploits matching and coloring procedures to plan the distribution of slots and channels across the entire network topology graph. The core contribution of this paper has hence been to derive some fundamental bounds on the minimum number of slots needed to deliver traffic to a root node, and verify that the prior introduced TASA algorithm performs at or close to these bounds.

Effectively, we were able to derive simple bounds by functionally relating the number of nodes, topology and data traffic to the system's duty-cycle. This, in turn, allows a proper dimensioning of industrial sensor IoT applications with stringent constraints in delay and power consumption. We were also able to corroborate that TASA performs very close to optimality, making it a viable protocol of choice for emerging applications. An important insight emerged when (fairly) comparing the new IEEE802.15.4e MAC with the legacy IEEE802.15.4/ZigBee MAC: The new MAC is up-to 80% more power efficient than the legacy one. This translates to significantly longer lifetimes of embedded sensor systems.

In summary, the TASA algorithm operates at optimal performance bounds and fulfills the requirements of many energy-constrained industrial applications. It is currently being discussed as a candidate scheduling solution in the newly formed IETF working group 6TSCH, which aims to link IEEE802.15.4e capabilities with prior IETF 6LoWPAN and ROLL standardization efforts. In this context, TASA will be integrated in the open-source implementation of the IoT protocol stack, developed within the OpenWSN project, in order to test its effectiveness in real scenarios.

Moreover, procedures and algorithms for exchanging TASA signaling messages, and for dealing with Packet Delivery Ratios smaller than 100% (i.e., link failure, topology changes) will be defined. Further future work will also be directed towards the definition of a schedule that provides QoS differentiations.

Acknowledgements

This publication was supported by the projects VITRO-257245, EXALTED-258512, and OUTSMART-285038, partially funded by the European Community, and by the PON projects ERMES-01-03113, DSS-01-02499, EURO6-01-02238, funded by the Italian MIUR.

REFERENCES

[1] General Electric, http://invent.ge/114PsAb, cEO Report, 2012.

[2] IEEE std. 802.15.4, *Part. 15.4: Wireless Medium Access Control (MAC) and PhysicalLayer (PHY) Specifications for Low-Rate Wireless PersonalArea Networks (LR-WPANs)*, IEEE standard for Information Technology, June 2011.

[3] J. Hui and P. Thubert, *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*, RFC 6282, IETF RFC 6282, September 2011.

[4] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, RFC 6550, IETF RFC 6550, March 2012.

[5] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, *Constrained Application Protocol (CoAP)*, IETF CoRE Working Group, February 2011.

[6] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack For The Internet Of (Important) Things," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1–18, December 2012.

[7] T. Watteyne, M. R. Palattella, and L. A. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals," IETF 6TSCH draft (work in progress), Febr. 2013.

[8] P. Thubert, T. Watteyne, M. R. Palattella, X. Vilajosana, and Q. Wang, "IETF 6TSCH: Combining IPv6 Connectivity with Industrial Performance," in *Proc. of Int. Workshop on Extending Seamlessly to the Internet ofThings (esIoT)*, Taiwan, July 2013.

[9] K. Pister and L. Doherty, "TSMP: Time synchronized mesh prootocol," in *Proc. of Int. Symp. Distributed Sensor Networks, DSN*, Florida, USA, Nov. 2008.

[10] HART Communication Protocol and Fundation, http://www.hartcomm2.org, Austin, TX, USA, 2012.

[11] IEEE std. 802.15.4e, *IEEE Standard for Local and metropolitan area networks – Part. 15.4:Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendament1: MAC sublayer*, IEEE Standards Association, Apr. 2012.

[12] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, "Traffic Aware Scheduling Algorithm for Multi-Hop IEEE 802.15.4eNetworks," in *Proc. of IEEE 23rd Int. Symposium on Personal Indoor and Mobile Radio Communications(PIMRC)*, Sydney, Australia, Sept. 2012.

[13] ——, "Traffic-Aware Time-Critical Scheduling In Heavily Duty-CycledIEEE802.15.4e For An Industrial IoT," in *Proc. of IEEE SENSORS*, Taipei, Taiwan, Oct. 2012.

[14] ISA, *ISA-100.11a-2009: Wireless Systems for Industrial Automation:Process Control and Related Applications*, NC, USA, 2009.

[15] M. Nixon, "A Comparison of WirelessHART and ISA100.11a," white paper, July 2012.

[16] S. Petersen and S. Carlsen, *WirelessHART versus ISA100.11a*, IEEE Industrial Electronics, Dec. 2011.

[17] T. Watteyne, A. Mehta, and K. Pister, "Reliability Through Frequency Diversity: Why Channel HoppingMakes Sense," in *Proc. of Performance Evaluation of Wireless Ad Hoc, Sensor, and UbiquitousNetworks (PE-WASUN)*, Oct. 2009.

[18] B. Kerkez, T. Watteyne, and M. Magliocco, "Fleasibility analysis of controller design for adaptive channel hopping," in *Proc. of ICST Int. Workshop on Performance Methodologies and Tools for WirelessSensor Networks, WSNPERF*, Oct. 2009.

[19] S. Yoon, R. Murawski, E. Ekici, P. Sangjoon, and Z. H. Mir, "Adaptive Channel Hopping for Interference Robust Wireless Sensor Networks," in *Communications (ICC), 2010 IEEE International Conference on*, may 2010, pp. 1 –5.

[20] G. Fiore, V. Ercoli, A. Isaksson, K. Landern, and M. DiBenedetto, "Multi-Hop Multi-Channel Scheduling for Wireless Control in WirelessHARTNetworks," in *Proc. of 14th IEEE Int. Conf. on Emerging Techonologies and Factory Automation*, Mallorca, Spain, Sept. 2009.

[21] A. Tinka, T. Watteyne, and K. Pister, "A Decentralized Scheduling Algorithm for Time Synchronized ChannelHopping," *Ad Hoc Networks*, vol. 49, no. 4, pp. 201–216, 2010.

[22] J. Tsitsiklis and K. Xu, "On the Power of (even a little) Centralization in Distributed Processing," in *Proc. of Int. Conf. on Measurement and Modeling (SIGMETRICS)*, Jun. 2011.

[23] DUST Networks: Intelligent Wireless Sensor Networks, http://www.linear.com/dust/, 2011.

[24] X. Ni and Y. Chen, "High Throughput MAC Scheduling Algorithm for Intelligent TransportationSensor Network," in *Proc. of Int. Conf. on Networking and Information Technology (ICINIT)*, Manila, Philippines, June 2010.

[25] J. Vasseur and J. Le Roux, *Path Computation Element (PCE) Communication Protocol (PCEP)*, RFC 5440, IETF RFC 5440, September 2009.

[26] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. W. Q.Wang, S. Glas, and K. Pister, "OpenWSN: A Standards-Based Low-Power Wireless DevelopmentEnvironment," *IEEE Transactions on Emerging Telecommunications Technologies*, Aug. 2012.

[27] Q. Wang, X. Vilajosana, and T. Watteyne, *6tus Adaptation Layer Specification. draft-wang-6tsch-6tus-00 (work in progress)*, IETF, March 2013.

[28] Q. Lampin, D. Barthel, and F. Valois, "Efficient Route Redundancy in DAG-based wireless sensor networks," in *Proc. of IEEE Wireless Communication and Networking Conference, WCNC*, Apr. 2010.

[29] O. Gnawali and P. Levis, *The Minimum Rank with Hysteresis Objective Function*, RFC 6719, Internet Engineering Task Force RFC 6719, September 2012.

[30] R. Diestel, *Graph Theory*, ser. Graduate Texts in Mathematics, S. Axler, F. Gehring, and P. Halmos, Eds. Springer, 1997.

[31] Di Marco, P. and Park, P. and Fischione, C. and Johansson, K.H., "Analytical Modelling of IEEE 802.15.4 for Multi-Hop Networks with HeterogeneousTraffic and Hidden Terminals," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, dec. 2010, pp. 1–6.

[32] *CC2430 Preliminary Data Sheet (rev. 2.1) SWRS036F*, Chipcon Products from Texas Instruments, Jun. 2007.