Traffic Aware Scheduling Algorithm for Reliable Low-Power Multi-Hop IEEE 802.15.4e Networks

Maria Rita Palattella*, Nicola Accettura†, Mischa Dohler‡, Luigi Alfredo Grieco† and Gennaro Boggia†
*SnT, University of Luxembourg (Luxembourg), maria-rita.palattella@uni.lu
†DEE, Politecnico di Bari (Italy), {n.accettura,a.grieco,g.boggia}@poliba.it
‡CTTC (Spain), mischa.dohler@cttc.es

***Abstract*—The Time Synchronized Channel Hopping (TSCH) protocol is part of the newly defined IEEE 802.15.4e standard and represents the latest generation of highly reliable low-power MAC protocols. With implementation details left open, we conceive here a novel Traffic Aware Scheduling Algorithm (TASA) by extending the theoretically well-established graph theory methods of matching and coloring by means of an innovative approach based on network topology and traffic load. TASA is able to support emerging industrial applications requiring low latency at low duty cycle and power consumption. Preliminary simulation results have also been reported to highlight the effectiveness of the proposed algorithm.**

I. Introduction

Supported by IEEE 802.15.4 standardization activities, wireless low-power and short-range communication technologies for consumer and industrial environments have been gaining in popularity in recent years. In its original formulation, the standard [1] supports various physical and Medium Access Control (MAC) mechanisms which ensure low power operation in a typically one-hop environment between a master node and several slave nodes. The current IEEE 802.15.4 MAC, however, is ill-suited for low-power multi-hop networks mainly because of (i) the high energy consumption due to relay/router nodes that have to be always active and (ii) the use of a single channel that implies impairments from interference and fading. For this reason, the IEEE 802.15e Working Group was created in 2008 to redesign the existing IEEE 802.15.4-2006 MAC standard towards a low-power multi-hop MAC, better suitable to emerging needs of embedded industrial applications.

The *Time Synchronized Channel Hopping* (TSCH) protocol, part of IEEE 802.15.4e standard [2] since 2010, is the latest generation of highly reliable and low-power MAC protocols. The initial concept emerged in 2006 as the proprietary Time Synchronized Mesh Protocol (TSMP) [3], conceived for multi-hop Wireless Sensor Networks (WSNs) based on the Highway Addressable Remote Transducer (HART) technology [4]. Through *time synchronization* and *channel hopping*, TSCH enables high reliability while maintaining very low duty cycles and thus utmost power efficiency.

In TSCH, nodes synchronize on a *slotframe* structure. A slotframe is a group of slots which repeats over time. Each node follows a schedule which tells it what it can do in every slot: transmit, receive, or sleep. For each active slot, the schedule indicates which channel to use and which neighbor to communicate with.

The IEEE 802.15.4e standard explains how the MAC layer executes a schedule, but it does not specify how such a schedule is built. Several schedule-based MACs and associated resource allocation algorithms have been suggested in the literature [5]–[8], but none of them can be directly applied to the problem at hand [9].

Whilst scheduling can either happen in a centralized or distributed fashion, centralized schedules are known to be superior to distributed ones for fairly static networks. Particularly under increasing load conditions, the performance can differ up to an order of magnitude [3], [10]. Recently, Tinka et al. [11] evaluated the use of a distributed approach in an IEEE 802.15.4e network, both by simulation and experimentally.

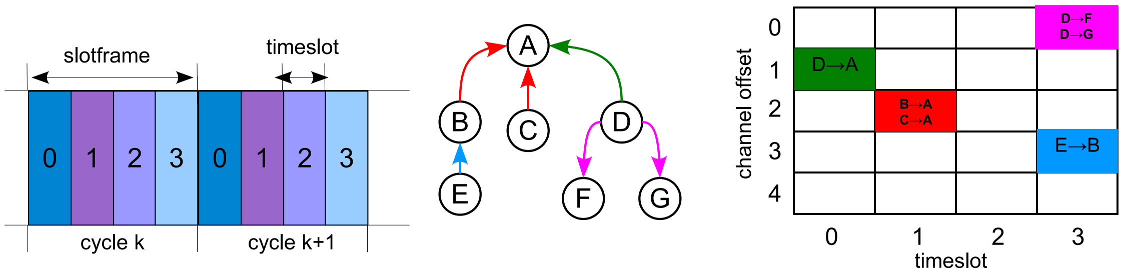
Instead, in this paper, we concentrate on a centralized approach where the schedule is built by a single node, typically the gateway giving global Internet connectivity. In contrast to prior art [5]–[8], the suggested Traffic Aware Scheduling Algorithm (TASA) builds time/frequency patterns based on the network topology <u>and</u> the traffic load at each node, whilst respecting low latency and low duty cycle at the same time. Since the schedule-based MACs mentioned above are not compliant to TSCH, we cannot compare them with TASA. Thus, we present a comparison between a TSCH-TASA based IEEE 802.15.4e network and a traditional IEEE 802.15.4 LR-WPAN.

The rest of the paper is organized as follows. In Sec. II, we detail the IEEE 802.15.4e TSCH protocol in order to aid the understanding of subsequent sections. In Sec. III, we present the analyzed network scenario, the notation used throughout the paper, and we detail the novel TASA scheme. Sec. IV reports simulation results highlighting the potentials of the proposed approach. Finally, Sec. V concludes the paper.

II. IEEE 802.15.4e TSCH: an overview

The TSCH protocol combines time slotted access, already defined in the IEEE 802.15.4 MAC protocol [1], with channel hopping. This technique has been proposed for a multi-hop network scenario where the spatial reuse of time and frequency is more challenging and allows for efficient use of available resources.

All nodes in the network are synchronized using a slotframe structure, as shown in Fig. 1(a). A single slot is long enough



(a) A 4-slot slotframe. (b) Dedicated and shared links. (c) Time/Frequency Schedule.

Fig. 1. A slotframe with 4-slots and possible schedule for a simple network topology with dedicated and shared links.

for the transmitter to send a maximum-length packet, and for the receiver to send back an acknowledgment. More node couples can exchange their frames at the same time (i.e., in the same slot) using different channel offsets. Thus, the use of several frequencies increases the network capacity. Moreover, channel hopping implies frequency diversity that mitigates the effects of interference and multipath fading. Finally, combined with slotted access, it also improves reliability.

Fig. 1(b) and Fig. 1(c) show an example topology and the associated schedule, respectively. Here, the slotframe is only 4 slots long, and there are 5 channel offsets. Each node in the network only cares about the cells it participates in. Note that, as depicted in Fig. 1(c), some cells are dedicated, while others can be shared between different links (e.g., $B \rightarrow A$ and $C \rightarrow A$). The IEEE 802.15.4e standard defines a simple backoff scheme for shared cells in case a collision occurs.

In the IEEE 802.15.4e TSCH, a *link* is defined as the pairwise assignment of a directed communication between devices in a specific slot within the slotframe, with a given channel offset [2]. Let (t, $chOf$) be the slot and the channel offset, respectively, assigned to a given link. The channel offset, *chOf*, is translated into a frequency f (i.e., a real channel) using the following translation function:

$$f = F\left\{(\text{ASN} + chOf) \mod n_{ch}\right\}, \tag{1}$$

where ASN = $(k \cdot S + t)$, is the *Absolute Slot Number* indicating the total number of slots that elapsed since the network was deployed, S is the slotframe size and k the slotframe cycle (Fig. 1). The function F is realized with a look-up-table containing the set of available channels. The value n_{ch} (i.e., the number of available physical frequencies) is the size of such a look-up-table. Moreover, the following constraints on t and $chOf$ hold: $0 \le t \le S-1$, and $0 \le chOf \le n_{ch}-1$. In an IEEE 802.15.4e network, 16 channels are typically available. Furthermore, a blacklist can be used to restrict the set of allowed channels for coexistence purposes. If the slotframe size, S, and the number of channel offsets, n_{ch}, are relatively prime, the translation function assures that each link rotates through k available channels over k slotframe cycles. In other words, successive frames over a same link are sent over different physical frequencies in successive k slotframe cycles.

As already specified, the IEEE 802.15.4e standard defines how the MAC executes a schedule, but it does not specify how to built such a schedule.

In the present work, we propose the Traffic Aware Scheduling Algorithm that builds a schedule, based on the network topology and the traffic load generated by each node in the network. In detail, we focus on a centralized approach since, as explained before, it is more workable in practice.

III. Traffic Aware Scheduling Algorithm

A. Network Model

We consider 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 master node (i.e., the PAN coordinator), and n_i, with $1 \le i \le N-1$, is the generic i-th node of the network (FFD or RFD device using IEEE 802.15.4 definitions [1]).

Fig. 2 shows the graph G of an example network. For each node n_i we define: i) its parent node, p_i; ii) the set of its child nodes, $ch(n_i)$; and iii) its sub-tree $ST(n_i)$, composed by n_i itself and all the nodes connected to it through multi-hop paths.

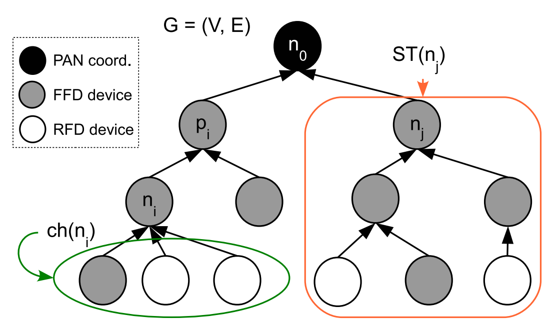


Fig. 2. Graph $G = (V, E)$ modeling a network with a tree topology.

Obviously, $ST(n_0)$ represents the whole network, G. Each node n_i is connected to its parent p_i with a dedicated link, $(n_i, p_i) \in E \subset V \times V$. Therefore, we have $|E| = N - 1$. We assume that all links are dedicated in order to avoid collision and to provide a reliable schedule. Note that every node n_i is connected only with its parent node p_i, even though it could have more neighbors and it could switch among them if there is the need.

B. Network Traffic

In emerging heterogeneous embedded applications, traffic is typically different between nodes but relatively constant over time. Thus, in the analyzed network scenario, we suppose that each node n_i in G, except the PAN coordinator, (i.e., n_i with $i \neq 0$), generates a constant integer number of packets, $\tilde{q}_i$, within a slotframe with a size equal to S slots. The network supports a multi-point-to-point traffic: all the packets are destined to the root node, n_0, and thus, each link (n_i, p_i) is used only in uplink. In fact, each source node n_i forwards its data frame to its parent node p_i in the tree rooted at the PAN coordinator.

Assuming that all $\tilde{q}_i$ packets generated by node n_i in a single slotframe are already available at the beginning of the slotframe, we can define the node *local queue level* in a given slot k as follows

$$q_i(k) = \begin{cases} \tilde{q}_i & \text{if } k = 0 \\ q_i(k-1) + 1\{rx_i(k-1)\} & \\ -1\{tx_i(k-1)\} & \text{if } 1 \leq k \leq S-1 \end{cases} \quad (2)$$

where $tx_i(k-1)$ and $rx_i(k-1)$ are events that are true if, during the time slot $k-1$, node n_i transmitted or received packets, respectively; moreover, $1\{X\}$ is the indicator function [12] equal to 1 only if the event X is true, otherwise it is 0.[1]

Let $\tilde{Q}$ be the network traffic load, i.e., the total number of packets delivered to the PAN coordinator within a single slotframe, which results in $\tilde{Q} = \sum_{i=1}^{N-1} \tilde{q}_i$.

Moreover, considering the sub-tree $ST(n_i)$, we define the *global queue level*, $Q_i(k)$, of a node in slot k as the total number of packets that are in the queues of the nodes belonging to the given sub-tree in that slot k. It is obviously given by

$$Q_i(k) = \sum_{j | n_j \in ST(n_i)} q_j(k). \quad (3)$$

Finally, given that a packet can move inside a sub-tree from node to node only during a single slot, it follows that the global queue level does not change in two consecutive slots, unless the node n_i itself (i.e., the root of the sub-tree $ST(n_i)$) transmits a packet [2].

C. Conflicts Definition

A node may interfere with another node in the network, so that they should not transmit at the same time on the same channel offset. As in prior works [6], [8], [13], in our model we assume that a *physical connectivity* graph $P = (V, C)$ is known a priori for the network G, where $C \subset V \times V$ is the set of edges such that $(n_i, n_j) \in C$ if either n_i and n_j can hear each other or one of them can interfere with a signal intended for the other node at the same time.

We now introduce some definitions that are useful for the description of our scheduling algorithm TASA. Note that a node n_i cannot transmit and receive at the same time, and it cannot receive from multiple nodes at the same time. To express this concept we define the set of *duplex-conflict* links, DC_i, for a given node n_i. It comprises the edges between node pairs in G that cannot transmit in the same slot reserved to the link (n_i, p_i). It is possible to distinguish two kinds of edges. First, if $n_j \in ch(p_i)$ with $j \neq i$, then all the edges (n_j, p_i) are in duplex-conflict with (n_i, p_i). In fact, if n_i and n_j transmit at the same time, the common parent would hear from both children and there should be a collision. Second, if $n_j \in ch(n_i)$, with $j \neq i$, the edges (n_j, n_i) are in duplex conflict with (n_i, p_i) because n_i cannot receive from its children, while it is transmitting to its parent. Obviously, only sets of *duplex-conflict free* links can be scheduled in the same slot k within the slotframe. Hereafter, we refer to the aforementioned set as $DCFL(k)$.

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.

A set of *duplex-conflict-free* links includes several subsets of *interference-conflict-free* links, each of them has to be scheduled on a different channel offset. In order to select these $ICFL_c(k)$ sets, with $1 \leq c \leq n_{ch}$, we will use an *Interference Conflict* Graph, $I(k)$. More details are provided in what follows.

D. The Scheduling Algorithm

As every centralized algorithm, TASA requires that the master node has complete topology information. Thus, we assume that the PAN coordinator knows the graph G, the physical connectivity graph P, and finally, the traffic load generated by each node, i.e., $\tilde{q}_i$, $\forall\, n_i \in V$, with $i \neq 0$[3].

Starting from the aforementioned information, the master node, n_0, can build the schedule, running the TASA procedure, summarized in Algorithm 1. Then, the scheduling will be forwarded to all the nodes in the network in order to let them know when they can transmit and on which channel.

The scheduling problem – Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA) – has been often formulated in literature as a graph (i.e., vertex or edge) coloring problem [13]–[15]. Instead, for resolving the problem at hand, i.e., to built a TSCH scheme, we propose a combination of *matching* and *vertex coloring* problems. In graph theory [16] a matching in a graph is a set of edges without common vertices, also called *independent* or *disjoint* edges. Whereas, a vertex coloring of a graph consists in assigning colors to each vertex of the graph such that two vertices sharing the same edge have different colors. In the present work, even though we follow the classical

[1] To understand Eq. (2), note that based on the IEEE 802.15.4e TSCH specifications, a slot is long enough for allowing the transmission of a single data frame. Therefore, in a single slot only one packet can be transmitted or received; moreover, events $tx_i(k)$ and $rx_i(k)$ cannot be both true in the same slot k.

[2] In other words, it results: $Q_i(k+1) = Q_i(k) - 1\{tx_i(k)\}$

[3] Both the procedure used for collecting the needed information at the master node and that for sending the schedule to all the nodes in the networks are practically viable [3], but beyond the scope of this paper.

approach, we propose an innovative technique for building both a matching and a vertex coloring, based on the network topology and the traffic load. Moreover, as explained later, we simplify the graph coloring problem that is known to be NP-complete [17].

Algorithm 1 Traffic Aware Scheduling Algorithm (TASA)

```
procedure SCHEDULING(G, P, q, Q, n_ch)        ▷ G: Graph
                                              ▷ P: PHY Connectivity Graph
                                              ▷ q: q_i(k) vector, Q: Q_i(k) vector,
                                              ▷ n_ch: number of available channels,
    k ← 0                                     ▷ slot initial value
    pattern ← [ ]                             ▷ time/frequency pattern initial value
    while q_0(k) ≠ Q̃ do                       ▷ The procedure is run until a
                                              ▷ pattern that allows to delivery to
                                              ▷ the PAN Coordinator all the packets
                                              ▷ generated in one slotframe, is built
        DCFL(k) ← MATCHING(G, q, Q, n_0, k)
        I(k) ← FINDINTERFGRAPH(DCFL(k), P)
                                              ▷ Building the Interference
                                              ▷ Conflict graph for DCFL(k)
        colored ← COLORING(I(k), Q)
        selected ← [ ]
        for n ← 1 to n_ch do
            selected ← GETFIRST(colored)
        end for
        UPDATE(q, Q, selected)                ▷ Updating q,Q
                                              ▷ according to links scheduled at k
        pattern ← pattern + [(k, selected)]
        k ← k + 1
    end while
    return pattern                            ▷ returning time/frequency pattern
end procedure
```

As shown in Algorithm 1, TASA includes two main procedures: i) *matching* and ii) *coloring* that are implemented in an iterative way. Step by step, the set of links that can be scheduled in each slot on the available channel offsets are selected. At the end of each step, based on the schedule built for the considered slot, local and global queue levels are updated, according to Eqs. (2) and (3). The update assures that the links to be scheduled in the next slot will be chosen according to the traffic load that each device has still to deliver to its parent node.

1) MATCHING: The first step of the proposed TASA algorithm consists in selecting a $DCFL(k)$ set to be scheduled in a given slot k.

This is equivalent to find a matching in G, i.e., a set of edges without common vertices. Different criteria can be used for finding a matching in a graph [16]. In the present work, the sets of *duplex-conflict-free* links are selected based on the traffic load supported by each link, in particular on the local and global queue levels.

In details, a *select function* is applied in a recursive fashion, crossing the tree from the root to the leaves. This function, applied to a parent node p_i, selects among its children, the child node n_i for which it results

$$Q_i(k) = \max\{Q_j(k) \mid n_j \in ch(p_i) \wedge \ q_j(k) \neq 0\}. \qquad (4)$$

Eq. (4) states that a slot k is reserved to a link (n_i, p_i) only when the child node n_i has at least a packet to transmit to its parent p_i 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.

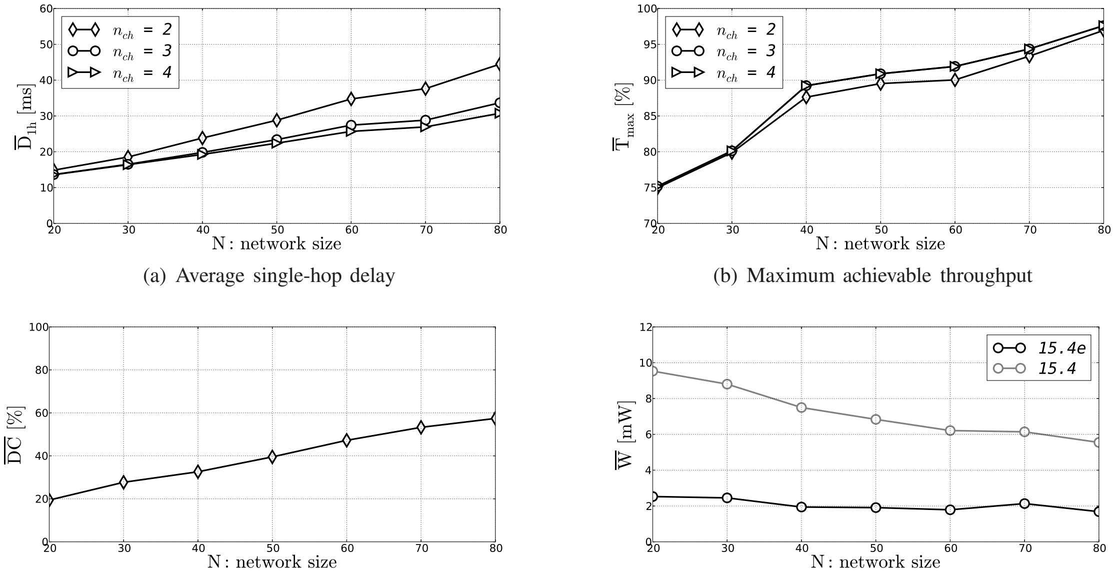
2) COLORING: The second step of our TASA procedure consists in assigning a channel offset to each link belonging to the *duplex-conflict-free* links set, $DCFL(k)$, selected for the slot k, in the previous step of the algorithm. To exploit the frequency spatial reuse, it is necessary to establish which links do not interfere and thus, can transmit on the same channel offset c, at the same time.

The frequency assignment problem can be formulated as a vertex-coloring problem of an interference graph [16]. For this reason, once the $DCFL(k)$ set has been chosen, we build a corresponding *Interference Conflict* graph $I(k) = \{V_I(k), E_I(k)\}$. We have that $V_I(k) \subset V$ is the set of transmitting nodes n_i of the dedicated links (n_i, p_i) belonging to $DCFL(k)$. Instead, $E_I(k) \subset C$ is the set of interfering links. Note that each link is bi-directional; in fact, a node can transmit a single data frame and receive its related ACK, during a single slot. Thus, it follows that two nodes n_i and n_j belonging to $V_I(k)$ interfere (i.e., $(n_i, n_j) \in E_I(k)$), if at least one of the edge in the set $\{(n_i, n_j), (p_i, p_j), (n_i, p_j), (n_j, p_i)\}$, belongs to C.

It has to be noticed that solving the coloring problem for a subset $V_I(k)$ of nodes in the network, allows us to simplify its complexity that normally increases with the number of vertices (i.e., NP-hard). For coloring $I(k)$ such that two nodes n_i and n_j are assigned different colors if $(n_i, n_j) \in E_I(k)$, we use an heuristic incremental method[4].

In detail, first of all the nodes n_i in $V_I(k)$ are ordered according to $Q_i(k)$ increase reversed order. 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, and those that interfere with n_i. The next vertex to be colored with the same color c is the first in the list of non-interfering nodes. 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.

[4]In heuristic incremental methods vertices are colored sequentially with the color chosen in response to colors already assigned in the vertex neighborhood. Each method varies in how the next vertex is selected and a new color assigned.



(a) Average single-hop delay

(b) Maximum achievable throughput

(c) Average network duty cycle

(d) Average power consumption.

Fig. 3. Performance of networks implementing the TASA algorithm.

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, it is possible to built the other set of *interference-conflict-free* links to be scheduled on different channels. The number of available channel offset, n_{ch}, could be smaller than the number of colors necessary for coloring $I(k)$. In this case, the slot k will be assigned to only some of the links in $DCFL(k)$; the other ones will be scheduled in the next slots.

IV. Performance Evaluation

To show the effectiveness of the TASA algorithm a simulator has been developed in Python language. We have implemented the algorithm in networks with a tree topology, deployed in an area of 200 $\times$ 200 m^2. Each node, located in a random position in the network, has a coverage range $R = 50$ m and a neighborhood cardinality that varies in the range [2, 20]. Since the scheduling in TASA is built according to the network topology, we analyze the performance in several scenarios, varying the total number of nodes, $N \in [20, 80]$.

Every node in the network generates on average 5 data frames within a single slotframe addressed to the PAN coordinator. The number of generated packets is uniformly distributed in the range [1, 9]. We assume that the slotframe has a fixed size of $S = 720$ slots and the duration of a single slot is 4 ms[5]. Finally, we consider a number of available channel offsets $n_{ch} \in [2, 16]$.

[5]Such duration allows the transmission of a data frame and the reception of an ACK whose sizes L and L_{ack} are respectively equal to 7 and 2 *aUnitBackoffPeriod* [1], as assumed in many prior works related to IEEE 802.15.4 standard. We have made this choice in order to compare the performance of the same network when it uses either the IEEE 802.15.4e TSCH MAC or the IEEE 802.15.4 MAC.

We have analyzed the performance of a network implementing TASA in terms of average single-hop delay, maximum achievable throughput, and network duty cycle. First of all, we have noted that simulation results do not change significantly for $n_{ch} \geq 4$, which shows the efficiency of the TASA scheme in terms of bandwidth utilization. Hence, we show only results for $n_{ch} \in [2, 4]$.

Fig. 3(a) shows the average single-hop delay, $\overline{D}_{1h}$, computed as the time elapsed from the moment a packet reaches the head of the queue, q_i, to the moment it is transmitted on the link (n_i, p_i). We can see that it decreases with increasing values of n_{ch} because they allow to better exploit the time-spatial reuse.

Moreover, we have computed the maximum achievable throughput, $\overline{T}_{max}$, under the considered operative condition, as the ratio between the total number of packets transmitted to the PAN Coordinator within a slotframe, $\tilde{Q}$, and the length of the schedule built with TASA. As we can see in Fig. 3(b), a higher throughput can be achieved in network with larger N, such as those used in large-scale energy-constrained urban monitoring application. Moreover, in order to further increase $\overline{T}_{max}$ more than two channel offsets have to be used. In fact, under these conditions, the TASA algorithm can work in a more efficient way, exploiting both time and frequency spatial reuse.

The suggested TASA algorithm is also able to provide a low network duty cycle, $\overline{DC}$, computed as the ratio between the length of the schedule and the slotframe size, because it allocates the data transmissions as soon as possible starting from the beginning of the slotframe. In Fig. 3(c), $\overline{DC}$ is shown only for the case $n_{ch} = 4$, because we found it does not depend significantly on the number of available channels. In fact, it is mainly dependent on the network traffic load. As shown in Fig. 3(c), $\overline{DC}$ is smaller than 50% in network with $N < 60$. This means that the network can sleep and save energy for more than half of the slotframe duration, as aimed in the emerging *green networking*.

Finally, we have compared the performance of networks implementing the IEEE 802.15.4e TSCH MAC jointly with our TASA algorithm versus the traditional IEEE 802.15.4 MAC protocol, in term of average per-node power consumption, $\overline{W}$. In order to analyze the behavior of IEEE 802.15.4 networks, we have used the Markov chain model proposed in [18] for modeling the unslotted CSMA/CA mechanism [1], because 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)[6]. Fig. 3(d) shows the average per-node power consumption, defined as the average power spent for transmitting a data frame with maximum payload length, and receiving the relative ACK within the same timeslot. It is obviously function of its traffic load and indirectly of the network topology. In fact, each parent node has to forward the data frames received from its children. As a consequence, it does not depend on the particular schedule, nor on the number of available channel offsets. The use of the TASA algorithm implies $\overline{W} \approx 2\ mW$, for all network size N, as depicted[7] in Fig. 3(d). We can see that the use of the TASA algorithm in an IEEE 802.15.4e TSCH network implies a gain of about 80% over the same network running the IEEE 802.15.4 MAC, expecially for small network size, N. Therefore, our TASA algorithm follows the new trend of developing green networking, and at the same time it suits the requirements of the emerging Internet of Things (IoT).

V. Conclusion

Energy is a major design drivers in embedded M2M networks [19], [20] and has hence been a major focus of this work. After the introduction of the main features of the emerging IEEE 802.15.4e TSCH MAC, a centralized scheduling algorithm, referred to as TASA, has been proposed for optimizing the performance of low-power multi-hop networks. Such an algorithm properly exploits with an innovative approach matching and coloring procedures to plan the distribution of slots and channel offsets across the entire network topology graph. The effectiveness of the proposed approach has been demonstrated using preliminary simulation results. In future work, we will test the validity of TASA using also theoretic arguments (based on theorems and lemmas) and we will extend the suggested TASA scheme to more complex network scenarios (including several slotframes, uplink and downlink traffic, mesh topologies, and so on). Worth noting is that simulations confirm that the new IEEE 802.15.4e MAC protocol is up-to 80% more power efficient than the traditional IEEE 802.15.4 MAC protocol for comparable network topologies.

[6] In detail, we have set the MAC parameters as suggested in [18], according to the standard [1]. In particular, because we consider only dedicated links in our TASA scheme, we have assumed that no retransmissions are allowed in the IEEE 802.15.4 network, in order to perform a coherent comparison.

[7] For illustrative purpose, when computing $\overline{W}$ the values of the Chipcon 802.15.4-compliant RF transceiver CC2430 have been utilized.

Acknowledgements

This publication was supported by the projects VITRO-257245, EXALTED-258512, and OUTSMART-285038, partially funded by the European Community; and by the project ERMES PON 01_03113/3, funded by the Italian MIUR.

References

[1] IEEE std. 802.15.4, *Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE standard for Information Technology, Sep. 2006.

[2] IEEE std. 802.15.4e, *Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendament 1: MAC sublayer*, IEEE standard for Information Technology, April 2012.

[3] K. Pister and L. Doherty, "TSMP: Time Synchronized Mesh Prootocol," in *Proc. of Int. Symp. Distributed Sensor Networks (DSN)*, Florida, USA, Nov. 2008.

[4] HART Communication Protocol and Fundation, Available online: http://www.hartcomm2.org.

[5] K. Arisha, M. Youssef, and M. Younis, *System-Level Power Optimization for Wireless Multimedia Communication*. Springer, 2002, ch. Energy-Aware TDMA-Based MAC for Sensor Network, pp. 21 – 40.

[6] S. Coleri-Ergen and P. Varaiya, "PEDAMACS: Power Efficient and Delay Aware Medium ACcess protocol for Sensor networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 7, pp. 920–930, 2006.

[7] L. v. Hoesel and P. Havinga, "A Lightweight Medium Access (LMAC) protocol for Wireless Sensor Networks," in *Proc. of Int. Conf. on Networked Sensing Systems (INSS)*, Tokyo, Japan, 2004.

[8] O. D. Incel, S. Dulman, and P. Jansen, "Multi-channel support for dense wireless sensor networking," in *Proc. of 1st Eur. Conf. on Smart Sensing and Context (EuroSSC)*, the Netherlands, Oct. 2006.

[9] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On Optimal Scheduling in Duty-Cycled IoT Industrial Applications using the IEEE 802.15.4e MAC," journal paper, under development.

[10] J. Tsitsiklis and K. Xu, "On the power of (even a little) centralization in distributed processing," in *Proc. of Int. Conf. on Measurement and Modelling of Computer Systems (SIGMETRICS)*, California, Jun. 2011.

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

[12] R. Nelson, *Probability, Stochastic Processes, and Queueing Theory*. Springer-Verlag, 1995.

[13] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Networks*, vol. 16, no. 4, pp. 985–997, 2009, dOI: 10.1007/s11276-009-0183-0.

[14] Y. Wang, W. Wang, X.-Y. Li, and W.-Z. Song, "Interference-Aware Joint Routing and TDMA Link Scheduling for Static Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 12, pp. 1709 – 1726, Dec. 2008.

[15] L. Paradis and Q. Han, "TIGRA: TImely sensor data collection using distributed GRAph coloring," in *Proc. of 6th IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, Hong Kong, Mar. 2008.

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

[17] D. Zuckerman, "Linear degree extractors and the inapproximability of Max Clique and Chromatic Number," *Theory of Computing*, vol. 3, pp. 103 – 128, 2007, doi:10.4086/toc.2007.v003a006.

[18] P. Di Marco, P. Park, C. Fischione, and K. Johansson, "Analytical Modelling of IEEE 802.15.4 for Multi-hop Networks with Heterogeneous Traffic and Hidden Terminals," in *Proc. of IEEE Global Telecommunications Conference 2010, (Globecom)*, Miami, Florida, USA, Dec. 2010.

[19] F. Ishmanov, A. S. Malik, and S. W. Kim, "Energy consumption balancing (ecb) issues and mechanisms in wireless sensor networks (wsns): a comprehensive overview," *European Transactions on Telecommunications*, vol. 22, no. 4, pp. 151–167, 2011. [Online]. Available: http://dx.doi.org/10.1002/ett.1466

[20] U. Alvarado, A. Juanicorena, I. Adin, B. Sedano, I. Gutirrez, and J. de N, "Energy harvesting technologies for low-power electronics," *Transactions on Emerging Telecommunications Technologies*, pp. n/a–n/a, 2012. [Online]. Available: http://dx.doi.org/10.1002/ett.2529