ACM Transactions on Embedded Computing Systems, Vol. 15, No. 3, pp. 41:1-41:29, March 2016.

Bandwidth Optimization and Energy Management in Real-Time Wireless Networks

GIANLUCA FRANCHINO, GIORGIO BUTTAZZO, and MAURO MARINONI,
Scuola Superiore Sant'Anna Pisa, Italy

In embedded systems operated by battery and interacting with the environment, a fundamental issue is the enforcement of real-time and energy constraints to guarantee a desired lifetime with a given performance. A lot of research has focused on energy management at the communication level; however, not many authors considered both real-time and energy requirements in wireless communication systems.

This article proposes El-SMan, a power-aware framework working in combination with MAC layer communication protocols for maximizing battery lifetime in wireless networks of embedded systems with real-time constraints. Exploiting the flexibility in bandwidth requirements, El-SMan adapts stream parameters to balance performance versus energy consumption, taking both lifetime and message deadlines into account.

CCS Concepts: • **Networks** → **Network protocol design**; *Link-layer protocols*; *Sensor networks*; • **Computer systems organization** → **Embedded systems**; **Real-time systems;**

Additional Key Words and Phrases: Energy management, real-time embedded systems, media access control, adaptive resource management

ACM Reference Format:
Gianluca Franchino, Giorgio Buttazzo, and Mauro Marinoni. 2016. Bandwidth optimization and energy management in real-time wireless networks. ACM Trans. Embed. Comput. Syst. 15, 3, Article 41 (March 2016), 29 pages.
DOI: http://dx.doi.org/10.1145/2851498

1. INTRODUCTION

Embedded systems cover a wide spectrum of application domains, such as consumer electronics, biomedical systems, surveillance, industrial automation, automotive, and avionics systems. In particular, the technology evolution of sensor and networking devices fostered the development of new applications involving distributed computing systems, many of them deployed in wireless environments and exploiting the mobility and the ubiquity of components. In most cases, devices are battery operated, making energy-aware algorithms of paramount importance to prolong the system lifetime. For instance, consider a distributed monitoring system composed by tens or hundreds of embedded nodes deployed in a specific area to detect intrusions or abnormal situations. Each node is equipped with sensors to detect the events of interest, motors to move in the environment, processing units to analyze data and compute control actions, and wireless transceivers to coordinate with the other nodes and transmit data to the base station. On one hand, the interaction with the environment creates implicit timing constraints that have to be enforced on the node activities to achieve a desired performance. On the other hand, since nodes are powered by batteries, the available energy

---

Authors' addresses: G. Franchino, G. Buttazzo, and M. Marinoni, Scuola Superiore Sant'Anna, Via Moruzzi 1, 56124, Pisa, Italy; emails: {g.franchino, g.buttazzo, m.marinoni}@sssup.it.

DOI: http://dx.doi.org/10.1145/2851498

must be carefully managed to prolong the system lifetime long enough to complete the mission. To take both requirements into account, the system should reduce power consumption as much as possible to extend its lifetime, but should use enough power to satisfy timing constraints. Hence, the objective of the system is to minimize energy consumption while guaranteeing timing constraints. Unfortunately, real-time and energy requirements have contrasting objectives and cannot be considered separately during system design. In particular, meeting real-time constraints on sensory and control activities requires the use of a suitable computing platform, running fast enough to complete all the tasks within their deadlines. However, running with a high speed causes a higher power consumption, which reduces the node lifetime. Similarly, at the communication level, reducing the end-to-end delay in message delivery requires the nodes to have a large transmission range to guarantee a single-hop connection, but large transmission ranges consume more power, reducing the system autonomy.

To reduce power consumption as much as possible, energy should be saved at different architecture levels. At the application level, specific devices can be switched off when they are not used for a sufficiently long interval of time or configured at a proper operating low-power mode (if any). At the operating system level, suitable scheduling and resource management algorithms can be adopted to reduce the clock frequency or switch the processor to a low-power state depending on the specific architecture, still guaranteeing the feasibility of a task set [Xu et al. 2007; Chen and Kuo 2007; Zhao and Aydin 2009]. At the network level, the transmission power of each node can be set at the minimum level that guarantees a given degree of connectivity. Moreover, as transceivers have different working modes (e.g., transmitting, receiving, and sleep mode), a node can set its transceiver on sleep mode to save energy whenever it does not need to communicate.

The semiconductor market offers several single-chip transceivers suitable to build wireless nodes. These devices have some smart features that can be exploited to design energy-aware transmission protocols [*CC*2420 2016], [muRata 2016]. For instance, most transceivers provide:

—Received Signal Strength Indicator (RSSI), which is a value proportional to the strength of the received RF signal. It can give a greedy estimate of the distance from the source, if the transmission power is known.
—Different levels of transmission power. They can be exploited, in conjunction with the RSSI, to save energy, adapting the transmission power to the distance between source and sink nodes.
—Different operating modes, each characterized by a different level of power consumption. The most common are sleeping, receiving, and transmitting.

Following the classification proposed in Ye et al. [2004], the main sources of energy waste at the Medium Access Control (MAC) level are:

—*Collision*. If a packet is corrupted, it has to be re-sent; therefore, both the sender and the receiver have to consume additional energy to exchange the packet.
—*Overhearing*. This is the energy wasted by a node when listening to packets directed to other nodes.
—*Control packet overhead*. This is the energy used to send and receive control packets, or adding control characters to the packets.
—*Idle listening*. This is the energy wasted by a node while listening to receive possible traffic that is not sent.

Depending on the application, idle listening can be the main cause of energy waste, since nodes can stay for a long time without sending messages, as in the case of a sensor network where no events are generated in the environment. Besides the energy waste sources described earlier, this work also considers the energy consumed by a transceiver to switch between operating modes. Such an overhead becomes relevant when the time intervals in which a node goes to idle are too short. In fact, in this case, the energy needed to switch from an active mode (transmitting or receiving) to the sleep mode and vice versa could be greater than that needed to stay always active.

1.1. Contributions and Summary

This work proposes an energy model and a related energy management algorithm, called Elastic Stream Manager (El-SMan), for guaranteeing lifetime constraints in wireless networks of embedded systems with real-time requirements. In particular, El-SMan is an elastic energy-aware algorithm that operates in conjunction with MAC protocols. It exploits the available flexibility in bandwidth requirements by adapting stream parameters to balance performance versus energy consumption, taking both battery lifetime and message deadlines into account. The proposed method is designed for working with scheduling-based MAC protocols and for network traffic generated by real-time streams. Both theoretical and experimental results are reported in the article.

This article extends and completes a previous work presented in Franchino et al. [2010]. In particular, it extends the analysis of the proposed approach by showing its ability to provide a quasi-optimal solution in polynomial time. Moreover, it reports a set of experimental results showing both the accuracy of the energy model on predicting the energy consumption and the effectiveness of El-SMan on energy saving.

The rest of the article is organized as follows. Section 2 describes the related work. Section 3 introduces the problem formulation and describes both the power consumption and the message stream models. Section 4 illustrates the El-SMan algorithm and presents an applicative example. Section 5 analyzes the properties of the algorithm. Section 6 presents some simulation results that show the performance of El-SMan in maximizing the network bandwidth while guaranteeing the system lifetime. Section 7 reports the experimental results that demonstrate the ability of the proposed approach in predicting and reducing the energy consumption. Finally, Section 8 concludes the article and presents some future developments.

2. RELATED WORK

The issues of real-time communication and energy saving over wireless networks have received much attention during the past years. However, not many authors addressed both problems simultaneously. Adamou et al. [2001] proposed a MAC protocol (PARMAC) that provides a guaranteed bandwidth for real-time traffic while reducing the energy consumption. Time is divided into frames of fixed length. Each frame is divided into a Reservation Period (RP) and a Contention-Free Period (CFP). During the RP, nodes contend to reserve transmission windows in the CFP; in the CFP, nodes send data packets without contention during their reserved slots, and sleep when they do not have to transmit or receive traffic. The authors highlight that PARMAC minimizes the idle time and allows a node to sleep during a CFP. Moreover, since the collisions are not frequent, a node needs to exchange fewer packets to complete a transmission; thus, it needs less energy to deliver a message. However, in this work, the authors do not give a method to compute the maximum energy consumption or compare the energy performance with respect to other protocols.

Caccamo et al. [2002] proposed a cellular network architecture with a Medium Access Control (MAC) protocol based on the Earliest Deadline First (EDF) [Liu and Layland 1973]. Implicit prioritization is achieved by exploiting the periodic nature of the traffic in sensor networks. The authors analyze the capacity of the network using an implicit EDF scheme, where each node locally generates the same MAC schedule and accesses the channel without collisions. The implementation of this scheme requires clock synchronization among nodes contending a channel. Moreover, slotted reservation may lead to waste of bandwidth if nodes either fail or, due to the variable packet size, do not use their reserved slots entirely. To address these problems, Crenshaw et al. [2005] presented a new version of the protocol, called Robust Implicit EDF (RI-EDF), which does not require clock synchronization, providing bandwidth reclamation and robustness in the presence of certain classes of node failures. Furthermore, the authors presented a power-aware extension, based on the partition of nodes in *sources* and *sinks*. Sources are nodes that transmit data. Sinks are node interested in the actual data sent; they can also send data but they are still classified as sinks. This extension needs a couple of additions in the protocol. First, to denote whether a packet is from a sink or a source, a new field is added in the packet headers. Second, by means of a periodic beacon packet, a sink may optionally solicit data from the sources. The goal of this power-saving mechanism is to reduce the energy consumption as much as possible without considering the network lifetime.

Unlike the previous protocols, which are mainly focused on the MAC layer, SPEED [He et al. 2003] is a protocol designed for real-time communication in sensor networks, which defines the behavior rules for all layers of the communication stack. In particular, SPEED provides a soft (probabilistic) guarantee on the real-time traffic delivery, through a feedback control scheduling mechanism [Stankovic et al. 2001]. Energy-savings performance is analyzed comparing the SPEED routing algorithm with other routing solutions available in the literature. A further interesting work is the RT-Link protocol, proposed by Rowe et al. [2006], which is a time-synchronized link layer protocol that guarantees a predictable life time and a bounded end-to-end delay across multiple hops. Other works concerning real-time wireless communication can be found in Demarch and Becker [2007] and Sobral and Becker [2008].

Koubaa et al. [2007] analyzed the power efficiency and the timeliness of the IEEE 802.15.4 standard [IEEE 802.15.4 Std-2006 2006] under the Guaranteed Time Slots (GTSs) mechanism. They also proposed a method to select the protocol parameters to trade power efficiency with delay-bound guarantees provided by this technology. Anastasi et al. [2010] considered the problem of reliable and energy-efficient data communication in multihop Wireless Sensor Networks (WSNs) based on the IEEE 802.15.4 standard. In particular, they analyzed the network performance by varying the setting of MAC layer parameters. The authors showed how sleep/wakeup scheduling and MAC parameters affect the network performance. Di Francesco et al. [2011] proposed a cross-layer framework that guarantees reliability and power saving in IEEE 802.15.4/ZigBee networks. The framework considers the application requirements in terms of reliability, the network topology, and the traffic status to select the MAC layer parameters that minimize the energy consumption. Toscano and Bello [2012] considered the low-power mechanisms of IEEE 802.15.4/ZigBee and 6LoWPAN networks. The authors provided a comparative analysis of the impact of such mechanisms on the network performance, assessed through a set of experiments that highlight the differences between the protocols. The resulting analysis provides a methodology to correctly tune the parameters of both protocols. Franchino and Buttazzo [2012] proposed WBuST, a MAC real-time communication protocol in wireless embedded systems. The channel access mechanism of the protocol is based on periodic beacons and time budgets. WBuST supports both real-time and best-effort traffic in cluster-tree networks and is able to reduce energy consumption.

Hoa and Kim [2012] proposed a couple of joint routing and scheduling algorithms for WSN applications. The algorithms are based on direct graphs, lossy ratio of node links, and an integer linear programming problem with the objective of minimizing the energy consumption and the mean communication latency. The authors do not take into account deadline constraints that will be part of future work. An energy efficiency method to manage multiple access points (APs) for Wi-Fi networks was proposed by

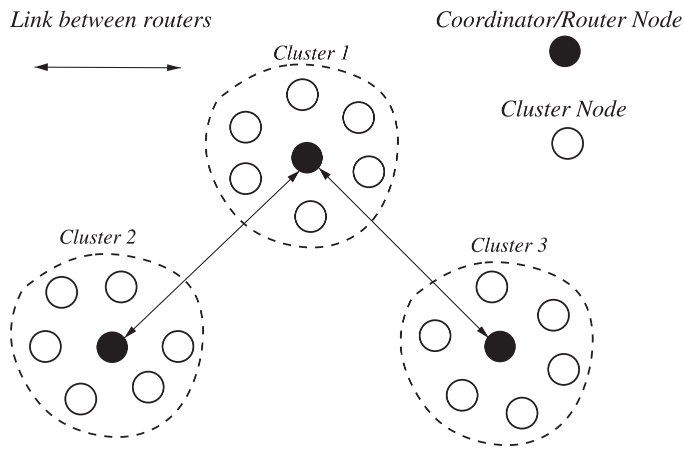


Fig. 1. Example of a clustered network of embedded devices.

Ma et al. [2012]. The authors proposed an energy-aware scheduling policy that reduces energy waste due to channel contention and guarantees the network performance in terms of throughput and fairness.

Note that, while the solutions discussed previously are tailored for a specific communication protocol, this article proposes a more general approach that can be applied to different MAC layer protocols based on Time Division Multiple Access (TDMA) scheduling algorithms. Also note that most of the existing solutions save energy by acting on the MAC layer protocol parameters, whereas this work operates at the stream level by selecting the proper stream parameters, assuming that protocol parameters are given and fixed. Moreover, while existing solutions are designed to minimize energy consumption, the objective of the proposed approach is to guarantee a target network lifetime by trading communication performance with energy consumption and maximizing the bandwidth of network streams.

3. PROBLEM AND MODELS

This work considers a real-time distributed system composed by a network of embedded devices (nodes) grouped into clusters/cells, as shown in Figure 1. Each cluster is formed by a master node, in charge of handling the communication among cluster nodes (acting as a cluster coordinator), and the communication with the masters of adjacent clusters (acting as a cluster router). Hence, the intercluster communication takes place between cluster routers. Moreover, it is assumed that all nodes in a cluster can directly communicate with the cluster coordinator/router. Nodes of the same cluster share a wireless channel, whose access is managed through a MAC layer protocol based on a TDMA scheduling algorithm.

It worth observing that, although a cluster-based architecture is not strictly necessary for the energy management framework proposed in this work, such a network topology is the most effective in practical applications where the channel access is regulated by a scheduling algorithm. In fact, in a fully distributed approach, the overhead for synchronizing the nodes and distributing the information needed to build the schedule can be significant, especially in dynamic environments where message streams can be added, removed, or changed at any time. In addition, fully distributed approaches are less stable and more error prone. For these reasons, it is usually preferred to create a group of nodes handled by a coordinator in charge of collecting the information and computing the schedule. Finally, note that a full connectivity in the cluster is not required, but it is sufficient that each node has a link with the cluster coordinator.

3.1. Network Traffic Model

In each cluster, the traffic generated by each node i is modeled by a message stream that can be of two kinds: real time (S_i^{RT}) or bandwidth guaranteed (S_i^{BG}). The former type of traffic is modeled by a periodic/sporadic real-time message stream described by three parameters (C_i, T_i, D_i), where:

—C_i is the maximum message size, measured in time units, of the messages generated by the stream.
—T_i is the interarrival period between consecutive messages in the stream. If the first message in stream S_i of node i is generated at time $t_{i,1}$, then the jth message in stream S_i will be generated at time $t_{i,j} = t_{i,1} + (j-1)T_i$, where $j = 1, 2, \ldots$.
—D_i is the relative deadline associated with messages in stream S_i, that is, the maximum amount of time that can elapse between a message arrival and the completion of its transmission.

Notice that the deadline of each message in stream S_i is only related to intracluster communication; that is, it is a local deadline and not an end-to-end deadline. Moreover, the time unit corresponds to the time needed to send a packet and receive its associated acknowledgment; hence, all stream and protocol parameters are expressed in number of packet transactions.

The bandwidth-guaranteed traffic is modeled by a stream described by a single parameter U_i^{BG}: the channel utilization, namely, the bandwidth required by the stream. This stream model does not consider messages subjected to a deadline, but it considers only traffic that needs to be served with a determined bandwidth.

For the sake of simplicity, it is assumed that each node i is associated with a message stream (S_i^{RT} or S_i^{BG}); however, it is not difficult to extend the analysis to the case in which a node is associated with more streams. Hence, the communication system is composed by a stream set $\Gamma = \{S_1, S_2, \ldots, S_n\}$, where each stream can be a real-time or a bandwidth-guaranteed stream.

The total demand produced by the stream message set is characterized by the total channel utilization, U, which is defined as

$$U = \sum_{i=1}^{n} U_i, \tag{1}$$

where $U_i = U_i^{RT} = C_i/T_i$ is the utilization of a real-time stream, while $U_i = U_i^{BG}$ is used for bandwidth-guaranteed streams.

While the hypothesis of constant parameters is applicable to several applications, the possibility to modify them at runtime could be used to adapt the system to different working scenarios. The possibility of varying the stream periods makes the communication system more flexible. For instance, in dynamic systems where communication streams can be created and destroyed at runtime, whenever a new stream cannot be accepted in the system because its utilization would overload the channel, instead of rejecting the stream, the system could try to reduce the other streams' utilizations, decreasing their channel occupation in order to accept the new request. A similar approach can be applied for bandwidth-guaranteed streams.

Elastic scheduling was first proposed in Buttazzo et al. [2002] for real-time tasks. Here, this framework is extended to message streams, considering each stream utilization as a spring with a given elastic coefficient, whose length can be modified by changing the maximum message duration C_i, the period T_i, or the guaranteed bandwidth U_i^{BG}.

Without loss of generality, in the rest of the article, it is assumed that each message stream, real time or bandwidth guaranteed, is described by three parameters: a minimum required bandwidth U_i^{min}, a maximum required bandwidth U_i^{max}, and an elastic coefficient ϵ_i, which specifies the flexibility of the stream to change its utilization.

The lower ϵ_i is, the lower the flexibility of stream S_i to reduce its utilization. In practical applications, more important streams are expected to have lower elastic coefficients, making them more resistant to adaptation. In the rest of the article, an elastic stream is denoted as $S_i = (U_i^{min}, U_i^{max}, \epsilon_i)$ and is characterized by an actual utilization U_i included in the range $[U_i^{min}, U_i^{max}]$.

3.2. Energy Model

This section presents the energy model describing the power consumption of the radio and its dependencies on the stream set parameters.

To compute the average power P_i consumed by node i, observe that each node:

—transmits for a certain amount of time depending on the associated stream parameters. For instance, considering a periodic stream, node i transmits for C_i time units every T_i. In other words, the time the node is on transmission is proportional to the stream bandwidth U_i^{RT}, or U_i^{BG} for bandwidth-guaranteed traffic;
—is in listening mode when the other nodes are transmitting;[1] and
—it goes on sleep during idle times.

To take switching overhead into account, it is assumed that each node takes t_i^{sw} units of time to switch between two operating modes and consumes a power P_i^{sw} during switching (for simplicity, it is assumed that both t_i^{sw} and P_i^{sw} are constant and independent of the modes).

Considering also those protocols where nodes exchange control packets to guarantee the network operation, it is assumed that a node i can make use of a channel bandwidth U_i^{ctrl} to transmit control messages.

The power consumed in transmission mode, receiving mode, and sleep mode is indicated as P_i^{tx}, P_i^{rx}, and P_i^{sl}, respectively. The average power consumed by node i can be written as a weighted sum of the powers consumed by the node in the different operating modes, considering that U_i is the fraction of time node i stays on transmission, $U - U_i$ is the fraction for receiving, U_i^{sw} is the fraction spent to switch between modes, U_i^{ctrl} is the fraction of time a node stays on transmission to deliver control information, $\sum_{j=1}^{n} U_j^{ctrl} - U_i^{ctrl} = U^{ctrl} - U_i^{ctrl}$ is the fraction for receiving control information from the other nodes, and $1 - U - U_i^{sw} - U_i^{ctrl}$ is the fraction for staying on sleep. So, the average power can be computed as

$$\begin{aligned} P_i \;=\; & U_i P_i^{tx} + (U - U_i) P_i^{rx} + U_i^{sw} P_i^{sw} \\ & + U_i^{ctrl} P_i^{tx} + \left(U^{ctrl} - U_i^{ctrl}\right) P_i^{rx} \\ & + \left(1 - U - U^{ctrl} - U_i^{sw}\right) P_i^{sl}. \end{aligned}$$

[1]Such a pessimistic assumption is made for the sake of simplicity, and will be relaxed later.

Observing that $U - U_i = \sum_{j=1, j\neq i}^{n} U_j$, the last equation can be rewritten as follows:

$$
\begin{aligned}
P_i &= U_i P_i^{tx} + \sum_{j=1, j\neq i}^{n} U_j P_i^{rx} + U_i^{sw} P_i^{sw} \\
&\quad + U_i^{ctrl}\left(P_i^{tx} - P_i^{rx}\right) + U^{ctrl} P_i^{rx} \\
&\quad + \left(1 - \sum_{j=1, j\neq i}^{n} U_j - U_i - U^{ctrl} - U_i^{sw}\right) P_i^{sl} \\
&= U_i P_i^{tx} + \sum_{j=1, j\neq i}^{n} U_j P_i^{rx} + U_i^{sw} P_i^{sw} + U^{ctrl} P_i^{rx} \\
&\quad + U_i^{ctrl}\left(P_i^{tx} - P_i^{rx}\right) + P_i^{sl} - \sum_{j=1, j\neq i}^{n} U_j P_i^{sl} \\
&\quad - U_i P_i^{sl} - U^{ctrl} P_i^{sl} - U_i^{sw} P_i^{sl},
\end{aligned}
$$

and gathering the terms

$$
\begin{aligned}
P_i &= U_i\left(P_i^{tx} - P_i^{sl}\right) + \left(\sum_{j=1, j\neq i}^{n} U_j\right)\left(P_i^{rx} - P_i^{sl}\right) \\
&\quad + U_i^{ctrl}\left(P_i^{tx} - P_i^{rx}\right) + P_i^{sl} + U^{ctrl}\left(P_i^{rx} - P_i^{sl}\right) \\
&\quad + U_i^{sw}\left(P_i^{sw} - P_i^{sl}\right).
\end{aligned} \tag{2}
$$

Now, let us define

$$
\begin{cases}
\Delta_i = U_i^{ctrl}\left(P_i^{tx} - P_i^{rx}\right) + P_i^{sl} + U^{ctrl}\left(P_i^{rx} - P_i^{sl}\right) \\
\qquad + U_i^{sw}\left(P_i^{sw} - P_i^{sl}\right) \\
P_i^{ts} = P_i^{tx} - P_i^{sl} \\
P_i^{rs} = P_i^{rx} - P_i^{sl}.
\end{cases} \tag{3}
$$

Hence, Equation (2) can be written as

$$
P_i = U_i P_i^{ts} + \sum_{j=1, j\neq i}^{n} U_j P_i^{rs} + \Delta_i. \tag{4}
$$

From Equation (4), it is clear that the power consumed by node i depends on the node utilization U_i, on the sum of the utilizations of the other nodes, on U_i^{sw}, U^{ctrl}, and U_i^{ctrl}.

Notice that the higher the power consumed in sleep mode, the lower P_i^{ts} and P_i^{rs} are, that is, the lower the contribution given by the node utilizations on the power consumption. The worst case happens when $P_i^{tx} \simeq P_i^{rx} \simeq P_i^{sl}$, and then $P_i = P_i^{sl} + U_i^{sw}(P_i^{sw} - P_i^{sl})$; that is, the node is uselessly wasting power to switch between operating modes.

If t is the time elapsed from the system start-up, let $P_i(t)$ be the power dissipated by node i at a time t; for a sufficiently large value of t, the energy $E_i(t)$ consumed in the worst case by node i at time t is

$$
E_i(t) = \int_0^t P_i(t)dt \simeq P_i \cdot t. \tag{5}
$$

The network lifetime is the time span from the system start-up to the instant when the network is considered nonworking. However, when a network is considered nonworking depends on the application [Chen and Zhao 2005; Blough and Santi 2002].

Under the same assumption considered in Equation (5), this work defines the network lifetime as the time at which the first node fails due to energy exhaustion. More precisely, if E_i^0 is the energy available at node i at time $t = 0$ and L_i is its lifetime, the network lifetime L_{net} is defined as

$$L_{net} = \min_i(L_i) = \min_i\left(\frac{E_i^0}{P_i}\right). \tag{6}$$

Analyzing the energy model expressed by Equation (4), it should be clear that it is very pessimistic, because it assumes that a node remains in a receiving state also when a message is sent to another destination. In this case, the energy is wasted due to *overhearing* (see Section 1). To reduce such an energy waste, a node that starts receiving a message directed to a different destination could extract the message length from the header and could switch to sleep for the message duration. It turns out that it is difficult to take such an energy into account in the energy model, as in several applications a node cannot know in advance when a stream does not send traffic to it. However, if such information is available, the node can better estimate its energy consumption reducing the energy wasted for overhearing. For instance, if $\widehat{\mathbf{S}}$ is the set of streams that do not send messages to node i, such a node can improve its energy model as follows:

$$P_i = U_i P_i^{ts} + \sum_{j=1, j\neq i, S_j \notin \widehat{S}}^{n} U_j P_i^{rs} + \Delta_i. \tag{7}$$

This example considers that stream $S_j \in \widehat{S}$ either always sends messages to node i or never sends messages to node i; therefore, its utilization U_j is not considered in the energy model. This is the same as considering $U_j = 0$. It is worth pointing out that, in several cases, a stream S_j can only send some messages to node i, so the utilization U_j has to be recomputed accordingly.

As stated before, to reduce the energy wasted due to the *idle listening,* a node can go in sleep state during idle times and when a message is not addressed to it. Notice that the way the nodes can exploit the idle time, to go in sleep state, depends on the MAC protocol. Moreover, as the idle time could be fragmented into small intervals, and switching between operative modes consumes energy, a node goes in sleep state only if the idle interval is sufficiently long to save energy. Formally, if t_{id} is the idle fragment length and remembering that $P_i^{rs} = P_i^{rx} - P_i^{sl}$, a node i goes on sleep if and only if

$$P_i^{rs}(t_{id} - 2t_{sw}) \geq 2P_i^{sw}t_{sw}$$

$$P_i^{rs} \geq P_i^{sw}\frac{2t_{sw}}{t_{id} - 2t_{sw}}. \tag{8}$$

The same reasoning applies when a node goes in sleep state because it is receiving a message sent to another destination. In this case, it is sufficient to substitute t_{id} with the message length in the last inequality.

Summarizing, the sources of energy waste discussed in Section 1 are taken into account by the energy model as follows. The algorithm assumes that the nodes switch to sleep mode during the idle time to reduce the energy due to *Idle listening*. The energy consumed by a node because of *Control packet overhead* is taken into account in the U_i^{ctrl} and U^{ctrl} parameters. The energy waste due to *Overhearing* is avoided since nodes know in advance the destination of each message. Finally, the energy waste due to *Collision* is avoided because the proposed algorithm is based on scheduling-based MAC protocols.

3.3. Problem Formulation

The following assumptions are considered while designing the algorithm.

At any time, each node is able to monitor its available energy. To save its energy, each node i transmits with a power P_i^{tx}, which is the minimum level that guarantees the network connectivity.

On the MAC layer, the communication protocol uses a TDMA scheme to manage the wireless channel. The advantage of this scheme with respect to the Carrier Sensing Multiple Access (CSMA) is that nodes do not waste energy to contend the channel access and there are no packet collisions.

In real-time communication systems, a message stream set Γ is said to be feasible if every message in each stream is sent by its deadline.

Several feasibility tests for message stream sets are based on the worst-case achievable utilization (WCAU) of the MAC protocol used to schedule the messages. In general, the WCAU of a scheduling protocol **A** represents the largest utilization $U^*(\mathbf{A})$ of the network such that, for any real-time stream set whose total network utilization is $U \leq U^*(\mathbf{A})$, **A** can guarantee the timeliness of each single real-time message.

To simplify the notation, when it is not necessary to specify the protocol, U^* is used instead of $U^*(\mathbf{A})$.

Notice that $U \leq U^*$ is generally a sufficient but not necessary condition for the schedulability of Γ.

In a system with a limited amount of energy, meeting all deadlines is not sufficient to guarantee the application goal. To take energy into account, the definition of feasibility is extended by taking into account the system lifetime.

Definition 3.1. Given a stream set $\Gamma = (S_1, S_2, \ldots, S_n)$ and a desired lifetime L_{net}^d, a schedule of Γ is said to be feasible if the message deadlines are met and the system lifetime L_{net} is greater than or equal to L_{net}^d.

From Equation (4), it can be observed that to reduce the energy consumption, besides reducing the transmission powers P_i^{tx}, it is possible to increase the sleep time by reducing the utilization of each node. In accordance with the elastic model presented in Section 3.1, it is assumed that the utilizations U_i can range in the interval $[U_i^{min}, U_i^{max}]$.

To summarize, our system can be described as follows:

—An elastic stream set $\Gamma = (S_1, S_2, \ldots, S_n)$
—The desired network lifetime L_{net}^d
—The WCAU U^* of the MAC protocol
—The vector of the initial available energies $\overrightarrow{E}^0 = (E_1^0, E_2^0, \ldots, E_n^0)$
—The transmission power vector $\overrightarrow{P}^{tx} = (P_1^{tx}, P_2^{tx}, \ldots, P_n^{tx})$
—The receiving power vector $\overrightarrow{P}^{rx} = (P_1^{rx}, P_2^{rx}, \ldots, P_n^{rx})$
—The sleep power vector $\overrightarrow{P}^{sl} = (P_1^{sl}, P_2^{sl}, \ldots, P_n^{sl})$
—The switching power vector $\overrightarrow{P}^{sw} = (P_1^{sw}, P_2^{sw}, \ldots, P_n^{sw})$

The objective is to maximize the total channel utilization U, maintaining $L_{net} \geq L_{net}^d$. In other words, the objective is to solve the following optimization problem:

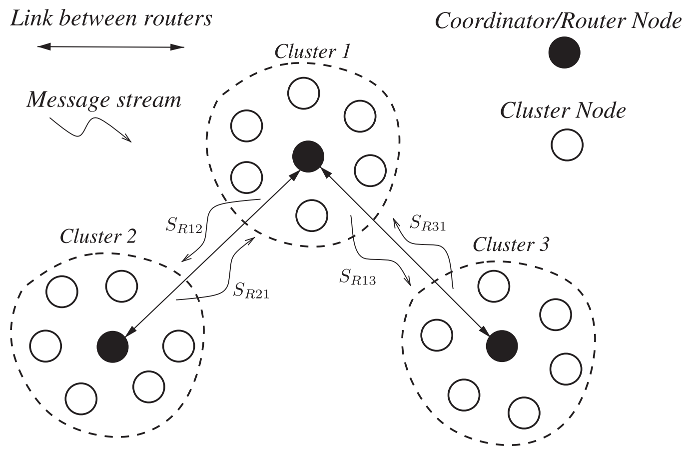


Fig. 2. Example of intercluster communication.

$$
\begin{aligned}
\max: \quad & U = \textstyle\sum_{i=1}^{n} U_i \\
s.t.: \quad & P_i \le \frac{E_i^0}{L_{net}^d} \qquad \forall i = 1, \ldots, n \\
& U_i^{min} \le U_i \le U_i^{max} \quad \forall i = 1, \ldots, n \\
& U \le U^*.
\end{aligned} \tag{9}
$$

3.4. Intercluster Communication Support

In the following, it is shown how to apply the energy model defined by Equation (4) and the associated optimization problem (Equation (9)) to the case of multihop communication among a set of clusters. Figure 2 shows a simple example of a network composed by three clusters forming a cluster-tree topology. The figure also displays the message streams representing the intercluster traffic exchanged among routers. In particular, stream S_{R12} represents the messages sent by the router of *Cluster 1* to the router of *Cluster 2*, and stream S_{R21} represents the messages sent by the router of *Cluster 2* to the router of *Cluster 1*. In the same way, streams S_{R13} and S_{R31} represent the traffic exchanged between *Cluster 1* and *Cluster 3*. It is worth noting that these intercluster streams can be represented by the same model used for intracluster traffic; that is, they can be described by the maximum message length, the interarrival period, and the one-hop deadline. Therefore, it is straightforward to apply the same energy management problem to a multihop scenario: it is sufficient to add the intercluster streams to the cluster stream sets. In the example of Figure 2, it is sufficient to add both streams S_{R12} and S_{R21} to the stream sets of *Cluster 1* and *Cluster 2*, and to add both streams S_{R13} and S_{R31} to the stream sets of *Cluster 1* and *Cluster 3*. Finally, the optimization problem (Equation (9)) can be separately applied to each cluster to maximize the cluster's stream set utilization while guaranteeing the desired lifetime of each cluster as defined by Equation (6). Note that the desired lifetime can be equal for each cluster, or a different lifetime can be selected for each cluster. In the rest of the article, the desired lifetime is considered equal for each cluster and defined by L_{net}^d.

Since each intercluster stream belongs to two cluster stream sets, one is the stream set of the cluster containing the router that generates the intercluster messages, and the other is the stream set of the cluster containing the router receiving the intercluster messages. For instance, streams S_{R12} and S_{R21} belong to both stream sets of *Cluster 1* and *Cluster 2*. Moreover, since the optimization problem (Equation (9)) is applied to both clusters, the solutions in the clusters can provide different utilizations. In this case, the utilization that guarantees the lifetime of both routers must be chosen. In other words, for P_{R1}, the average power dissipated by the router of *Cluster 1*, and P_{R2}, the average power dissipated by the router of *Cluster 2*, the selected utilization U_{R12} must guarantee that both $P_{R1} \leq E^0_{R1}/L^d_{net}$ and $P_{R2} \leq E^0_{R2}/L^d_{net}$. If a router node consumes more energy when in transmission mode than in receiving mode, then the utilization that guarantees the lifetime of both routers is the smallest one among the two solutions. Vice versa, if the node consumes more energy when in receiving mode, the utilization that guarantees the lifetime of both routers is the greatest one among the two solutions. This last observation comes from the results provided by Lemma 5.2 in Section 5.1, which highlights the fact that the power dissipated by a node mostly depends on the amount of traffic it delivers if the transmission power is greater than the receiving power; instead, in the opposite case, it mostly depends on the amount of received traffic.

It is worth pointing out that, since the energy model (Equation (4)) and the associated optimization problem (Equation (9)) are based on the message stream utilizations and on the utilization of the control information, that is, U_i^{ctrl}, they do not depend on how the intracluster schedule is coordinated with the intercluster schedule. This issue only involves the underlying MAC layer protocol, which has to rule the channel access. By the same token, the relation between per-cluster deadlines and end-to-end deadlines is not an issue in this work. Moreover, the intercluster streams are assumed independent of the intracluster streams. For these reasons, the channel access schedule, end-to-end deadlines, and multihop communication issues are not further discussed. A reader interested in these topics can refer to Caccamo et al. [2002], Koubaa et al. [2006], and Franchino and Buttazzo [2012].

4. THE ALGORITHM

The algorithm for solving the problem is based on an extended concept of network overload that also considers the network lifetime. More specifically, the following definition is adopted in this article.

Definition 4.1. Given a stream set $\Gamma = (S_1, S_2, \ldots, S_n)$ with utilization U and lifetime L_{net}, the communication system is considered overloaded if $U > U^*$ or $L_{net} < L^d_{net}$.

4.1. The Elastic MAC Algorithm

The El-SMan algorithm starts by setting $U_i = U_i^{max}$ for all streams. Then, if the system is overloaded, the problem is to find a new channel utilization U^d such that $U^d \leq U^*$ and $L_{net} \geq L^d_{net}$.

If the overload is caused by $U > U^*$, the elastic algorithm proposed in Buttazzo et al. [2002] can be used to compress the utilizations U_i so that $U = U^d \leq U^*$. If lifetime is also met under this setting, the algorithm stops returning the utilization vector $\overrightarrow{U} = (U_1, U_2, \ldots, U_n)$. Otherwise, when $L_{net} < L^d_{net}$, if S_y is the stream with the shortest lifetime, all stream utilizations are reduced to increase L_y up to L^d_{net}. A sample scenario is illustrated in Figure 3, where, before compression, streams S_1 and S_4 cannot reach the desired lifetime and S_4 is the stream with the shortest lifetime less than L^d_{net}, which determines the network lifetime.

To guarantee the lifetime constraint of the network, the power consumption of the overloaded streams (S_1, S_4) has to be reduced. Considering the dependencies expressed in Equation (4), all the stream utilizations are properly reduced to reach the desired network lifetime. In particular, the stream utilizations must be reduced so that, for all i, $U_i^{min} \leq U_i \leq U_i^{max}$ and $L_4 = L^d_{net}$.

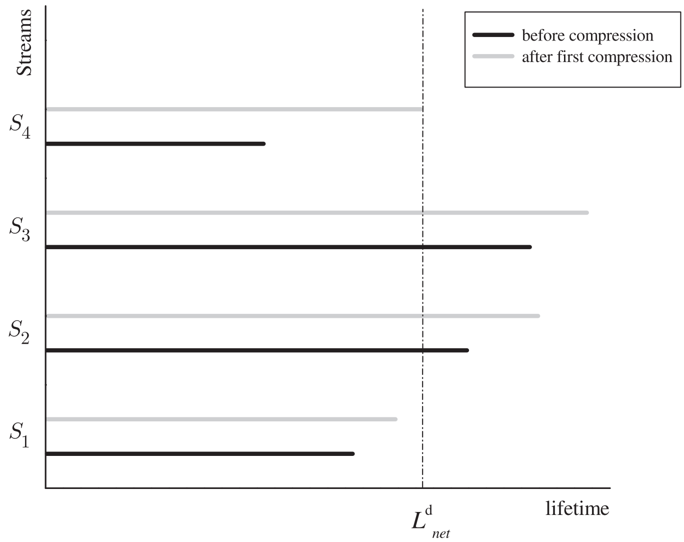


Fig. 3. Example on the nodes' lifetime.

Notice that, after such a compression, all stream lifetimes will increase, but the lifetime of some streams could still be less than L^d_{net}. For instance, in the example shown in Figure 3, stream lifetimes after the first compression are indicated by the gray bars, and lifetime of stream S_1 is still less than L^d_{net}. In this condition, the stream with the shortest lifetime is S_1; thus, a new compression must take place to bring $L_1 = L^d_{net}$. In general, the compression procedure has to be repeated until the network lifetime constraint is satisfied, that is, until $\forall i \, P_i \leq E^0_i / L^d_{net}$.

Notice that, at each compression step, stream utilizations are reduced by using the elastic algorithm presented in Buttazzo et al. [2002], according to which the new utilizations are computed as follows:

$$U_i = U_i^0 - \frac{U^0 - U^d}{\epsilon_{tot}} \epsilon_i, \tag{10}$$

where U^d is the total desired utilization, U^0_i is the nominal utilization of stream S_i, U^0 is the sum of nominal utilizations, and $\epsilon_{tot} = \sum_{i=1}^{n} \epsilon_i$. The way to choose the elastic coefficients is discussed in Section 5.

Intuitively, Equation (10) reduces each stream utilization proportionally to its relative elasticity ($\epsilon_i/\epsilon_{tot}$) and the exceeding total utilization ($U^0 - U^d$). In this way, streams with lower elasticity ϵ_i will be less affected by the algorithm.

In the classical elastic algorithm, the only constraint that U^d must satisfy is $U^d \leq U^*$. In this context, however, the lifetime constraint must also be met; hence, U^d must also satisfy $L_{net} \geq L^d_{net}$. Considering that the network lifetime is imposed by the stream (S_y) with the shortest lifetime ($L_{net} = L_y$), the nominal power P^0_y consumed by S_y must be decreased to reach the desired lifetime, that is, $P^d_y = E^0_y / L^d_{net}$.

In the following, the stream power expressed by Equation (4) is combined with Equation (10) to derive the value of U^d to be used as a reference in the compression algorithm.

From the energy model (Equation (4)):

$$\begin{aligned}\Delta P_y &= P_y^0 - P_y^d \\ &= \left(U_y^0 - U_y\right)P_y^{ts} + \sum_{j\neq y}^{n}\left(U_j^0 - U_j\right)P_y^{rs} \\ &\quad + \Delta_y^0 - \Delta_y. \end{aligned} \tag{11}$$

Using Equation (10), $(U_y^0 - U_y)$ and $(U_j^0 - U_j)$ can be substituted in Equation (11) as follows:

$$\begin{aligned}\Delta P_y &= \left(\frac{U^0 - U^d}{\epsilon_{tot}}\epsilon_y\right)P_y^{ts} + \sum_{j\neq y}^{n}\left(\frac{U^0 - U^d}{\epsilon_{tot}}\epsilon_j\right)P_y^{rs} \\ &\quad + \Delta_y^0 - \Delta_y \\ \Delta P_y &= \frac{U^0 - U^d}{\epsilon_{tot}}\left(\epsilon_y P_y^{ts} + \sum_{j\neq y}^{n}\epsilon_j P_y^{rs}\right) + \Delta_y^{old} - \Delta_y \\ \Delta P_y &= \frac{U^0 - U^d}{\epsilon_{tot}}\left(\epsilon_y P_y^{ts} + (\epsilon_{tot} - \epsilon_y)P_y^{rs}\right) + \Delta_y^0 - \Delta_y \\ \Delta P_y &= \frac{U^0 - U^d}{\epsilon_{tot}}\left(\epsilon_y P_y^{tr} + \epsilon_{tot} P_y^{rs}\right) + \Delta_y^0 - \Delta_y, \end{aligned} \tag{12}$$

where $P_y^{tr} = P_y^{ts} - P_y^{rs} = P_y^{tx} - P_y^{rx}$.

From Equation (12), it follows that

$$\frac{U^0 - U^d}{\epsilon_{tot}} = \frac{\Delta P_y - \left(\Delta_y^0 - \Delta_y\right)}{\left(\epsilon_y P_y^{tr} + \epsilon_{tot} P_y^{rs}\right)}. \tag{13}$$

Substituting Equation (13) into Equation (10), we have

$$U_i = U_i^0 - \frac{\Delta P_y - \left(\Delta_y^0 - \Delta_y\right)}{\epsilon_y P_y^{tr} + \epsilon_{tot} P_y^{rs}}\epsilon_i. \tag{14}$$

From the definition of Δ_i (Equation (3)) in Section 3.2, observe that Δ_y depends on the percentage of time spent to transmit and receive control messages, that is, U_y^{ctrl} and U^{ctrl}, and on the percentage of time U_y^{sw} spent to switch between working modes. Thus, since in general the amount of control information exchanged by the nodes does not depend on the stream utilizations, it is reasonable to assume that the value of stream utilizations has no effect on U_y^{ctrl} and U^{ctrl} and has little effect on U_y^{sw}. Moreover, it is also reasonable to assume that if the stream utilizations are reduced, U_y^{sw} is reduced as well, and hence, $\Delta_y^0 \geq \Delta_y$. Therefore, $\Delta_y^0 - \Delta_y$ can be assumed to be a small positive quantity and it can safely be neglected still leading to a conservative result. In this case, Equation (14) becomes

$$U_i = U_i^0 - \frac{\Delta P_y}{\epsilon_y P_y^{tr} + \epsilon_{tot} P_y^{rs}}\epsilon_i, \tag{15}$$

where ΔP_y is given by

$$\Delta P_y = P_y^0 - P_y^d = P_y^0 - \frac{E_y^0}{L_{net}^d}. \tag{16}$$

```
1  StreamCompress($L^d_{net}$){
2    for (each $i \in \Gamma$) {
3      $U_i^0 = U_i^{max}$;
4    }endfor
5    for (each $i \in \Gamma$) {
6      $P_i^0 = U_i^0 P_i^{ts} + \sum_{j \neq 1}^{n} U_j^0 P_i^{rs} + \Delta_i^0$;
7      $L_{net} = \min_i \left( \frac{E_i^0}{P_i} \right)$;
8    }endfor
9    while ($L_{net} < L^d_{net}$) {
10       $y =$GetMaxPowerIndex($\Gamma$);
11       $\Delta P_y = P_y^{old} - \frac{E_y^0}{L^d_{net}}$;
12       $U^d = U^0 - \frac{\Delta P_y \epsilon_{tot}}{\epsilon_y P_y^{tr} + \epsilon_{tot} P_y^{rs}}$;
13       feasible = ElasticCompress($\Gamma$, $U^d$);
14       if (feasible == FALSE) return INFEASIBLE;
15       for (each $i \in \Gamma$) {
16         $P_i = U_i P_i^{ts} + \sum_{j \neq 1}^{n} U_j P_i^{rs} + \Delta_i$;
17         $L_{net} = \min_i \left( \frac{E_i^0}{P_i} \right)$;
18       }endfor
19    }endwhile
20   return FEASIBLE;
21   }
```

Fig. 4. Stream compression algorithm.

The overall stream compression algorithm is shown in Figure 4. Summarizing, El-SMan works as follows. The input parameter is the desired lifetime L^d_{net}. The two loops from line 2 to line 8 set the initial stream utilizations and compute the system lifetime L_{net}. The cycle starting at line 9 and ending at line 19 is the core of El-SMan. First, the index of the stream with the lowest lifetime is obtained (line 10), and then the desired utilization U^d is computed according to Equation (13). U^d is passed to the elastic compression procedure defined in Buttazzo et al. [2002]. After that, the new system lifetime is calculated and the cycle is repeated until either the lifetime constraint is met ($L_{net} \geq L^d_{net}$) or the system results become infeasible ($\sum U_i^{min} > U^d$).

The complexity of the algorithm in Figure 4 is $O(n^3)$. In fact, in the worst case, all the n streams might be compressed using the elastic procedure, which has an $O(n^2)$ complexity.

4.2. Applicative Example

To clarify the proposed approach, this subsection describes how to compute the parameters of the energy model (Equation (4)) considering the RI-EDF protocol [Crenshaw et al. 2005] as an example. RI-EDF is a MAC layer protocol for cellular structured networks that makes use of a distributed EDF algorithm to schedule node messages. Under RI-EDF, the system time is measured in packets or frames; that is, for each stream S_i, the message length C_i, the period T_i, and the deadline D_i are expressed in number of packets. This assumption allows the use of a preemptive algorithm, in that any preemption is done at the packet level, which is considered an atomic unit.

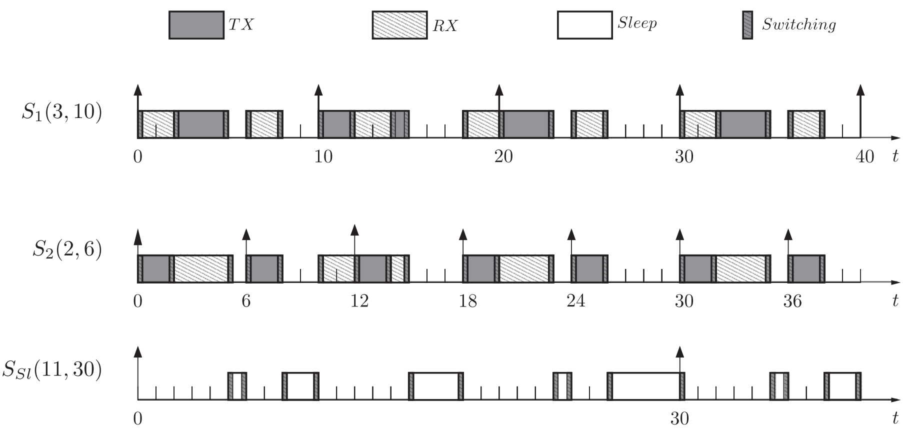


Fig. 5. U_i^{sw} calculation with RI-EDF.

Moreover, since each node computes the global channel schedule, any node knows how many packets it can transmit before being preempted by a higher-priority transmission.

Let us assume a network where nodes are grouped into hexagonal cells, each one working on a different radio channel. The channels are assigned such that cochannel interferences are avoided. Moreover, in each cell, any node can listen to any other. Without any loss of generality, it is assumed that each cluster/cell is composed by $n-1$ nodes plus a router node that is in charge of intercell communication. Each node, including the router, is associated with an elastic stream of periodic messages. For further details on the protocol, see the original papers [Caccamo et al. 2002; Crenshaw et al. 2005].

Observe the following:

—Node i is on transmission mode either when delivering its periodic messages, hence $U_i = C_i/T_i$, or when transmitting recovery messages.
—Each node knows the schedule of the communication, such that it can compute the bandwidth U_i^{sw} wasted switching between operating modes. U_i^{sw} can be computed by considering the total number N_i^{sw} of mode switches, performed by node i in the hyperperiod H (i.e., the least common multiple of the stream periods). Thus, it turns out that $U_i^{sw} = (N_i^{sw}/H)t_i^{sw}$.
—The bandwidth that each node i uses to deliver control messages, U_i^{ctrl}, depends on the number of recovery messages it has to transmit in the hyperperiod H. In detail, let t_{rec} be the time needed to transmit a recovery message, and N_i^{rec} be the number of recovery messages transmitted by node i during H time units; it follows that:

$$U_i^{ctrl} = \frac{N_i^{rec} t_{rec}}{H}.$$

The value of N_i^{rec} mainly depends on the channel status, for example, on the packet error rate, which varies during the network operation. Thus, the value of N_i^{rec} can be computed only statistically.

To complete the example, it remains to show how to calculate the value of U_i^{sw} for each stream S_i. Consider for simplicity a network composed by two nodes, associated with a stream $S_1(C_1 = 3, T_1 = 10)$ and $S_2(C_2 = 2, T_2 = 6)$, respectively. The schedule of the channel access by the two nodes is shown in Figure 5. The figure shows both the schedule of the transmission time of each node and the schedule of the sleep time represented by a dummy node associated with a stream $S_{Sl}(C_{sl}, T_{Sl})$. In particular, remembering that the network nodes go on sleep during the idle time, it turns out that C_{Sl} is equal to the idle time during the hyperperiod H and, consequently, T_{Sl} is equal to H, which in turn is equal to $LCM(T_1, T_2) = 30$. The sleep time, C_{Sl}, can be simply computed as follows:

$$C_{Sl} = H\left(1 - \sum_{i=1}^{2} U_i\right) = 30\left(1 - \frac{3}{10} + \frac{2}{6}\right) = 11.$$

Note that when node 1 goes on transmission, node 2 goes on receiving mode and vice versa. Furthermore, both nodes go on sleep when the dummy node, associated with stream S_{Sl}, "accesses the channel." From these last observations, note that $N_1^{sw} = N_2^{sw} = N^{sw}$. Moreover, to calculate N^{sw}, it is sufficient to count how many times each node, including the sleep node, accesses the channel during the hyperperiod $H = 30$. Figure 5 shows that node 1 accesses the channel 4 times, and both node 2 and the sleep node access it 5 times. Thus, it turns out that $N^{sw} = 14$ and $U_1^{sw} = U_2^{sw} = 14/30 = 0.467$.

Finally, notice that, as stated in Section 4.1, it is assumed that $\Delta_y^0 - \Delta_y = 0$; hence, with El-SMan, it is not strictly necessary to calculate U_i^{sw} and U_i^{ctrl}.

Although the example reported previously refers to RI-EDF, the proposed energy model and the related El-SMan framework is general enough to be applied to all the MAC protocols for which it is possible to derive the percentage of time in which a node is in transmission, receiving, and sleep mode. This is always possible for scheduling-based protocols, because the percentage of time spent in the different working modes is proportional to the stream utilizations. For instance, WBuST [Franchino and Buttazzo 2012] is a MAC protocol where the channel access is organized in periodic windows, delimited by the transmission of a periodic beacon, and composed by a group of time slots assigned to each node. If B_i is the time slot assigned to node i, B_i depends on the parameters of stream S_i. In particular, B_i is directly proportional to U_i, and the percentage of time that the node is in transmission mode is given by B_i divided by the communication window length, that is, the beacon period. Similarly, the percentage of time that a node is in receiving mode is given by the sum of the time slots assigned to the other nodes divided by the beacon period. Moreover, the protocol allocates a sleep interval at the end of each communication window; in this way it is also possible to compute the percentage of time that nodes spend in sleep mode. U_i^{sw}can be derived considering that in each communication window (superframe), a node switches from receiving to transmission at the beginning of its reserved slots and switches back to receiving mode at the end of the slots; it switches to sleep mode at the beginning of the sleep interval and switches to receiving mode at the end of the sleep mode to receive the beacon.

A further example is presented later to shown how El-SMan can be used on top of a protocol compliant with the IEEE 802.15.4 MAC layer. The parameters of the MAC protocol are computed by the method proposed by Koubaa et al. [2007] to achieve a tradeoff between the maximum transmission delay and the energy consumption. In particular, their method is used to derive a transmission delay bound as a function of the allocation of the GTSs and the Duty Cycle (DC), computed as the ratio of the Superframe Dimension (SD) and the Beacon Interval (BI). It is assumed that a node i transmits sporadic messages exploiting the GTS_i slot group, that is, the n_i slots reserved for the node during the CFP. It is also assumed that during the Contention Access Period (CAP), the nodes are on receiving mode. The structure of the beacon frame is illustrated in Figure 6. From this figure, it can be observed that a node i is on transmission mode during GTS_i for a maximum time equal to $n_i T_{data}$; it is on receiving

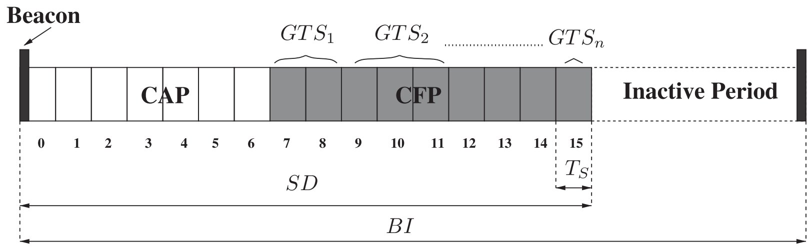


Fig. 6. IEEE 801.5.4 Beacon interval.

mode for the remaining part of the superframe and on sleeping mode during the Inactive Period. Note that, considering a packet transmission without acknowledgment, the length of a time slot T_S is equal to the packet transmission length T_{data} plus the InterFrame Spacing (IFS). Under this setting, the fraction of time that a node i spends in transmission mode depends on the number of slots n_i allocated in each BI, which depends on both the stream parameters and the allocation method. Hence, it turns out that in this case, $U_i = \frac{n_i T_{data}}{BI}$, and the resulting energy model is

$$P_i = U_i P_i^{tx} + \sum_{j=1, j\neq i}^{n} U_j P_i^{rx} + \frac{CAP + IFS \sum_i n_i}{BI} P_i^{rx} + (1 - DC) P_i^{sl}, \tag{17}$$

where

$$DC = \frac{SD}{BI} = \frac{CAP + IFS \sum_i n_i}{BI} + U_i + \sum_{j=1, j\neq i}^{n} U_j. \tag{18}$$

Substituting the expression of DC in Equation (17) and gathering the term, it follows that

$$P_i = U_i P_i^{ts} + \sum_{j=1, j\neq i}^{n} U_j P_i^{rs} + \Delta_i, \tag{19}$$

where

$$\Delta_i = \frac{CAP + IFS \sum_i n_i}{BI} P_i^{rs} + P_i^{sl}. \tag{20}$$

For the sake of simplicity, in Equation (19), U_i^{sw} is not considered and U_i^{ctrl} is taken into account in the term $(CAP + IFS \sum_i n_i)/BI$. However, U_i^{sw} can be derived following the same method explained previously for WBuST.

Given a stream set where the stream parameters C_i and $T_i = D_i$ are fixed, and given for each stream S_i the number of allocated slots n_i, through the method proposed by Koubaa et al. [2007], it is possible to compute for each S_i a transmission delay bound $D_{n_i,max}$ and the minimum Duty Cycle ($DC_{i,min}$), that is, the minimum energy consumption that guarantees a transmission delay not greater than $D_{n_i,max}$. Given a desired network lifetime L_{net}^d, if there exists a slot allocation such that for all i, $D_{n_i,max} \leq D_i$ and $L_{net} \geq L_{net}^d$, then the message deadlines and the network lifetime are both satisfied and it is not necessary to apply El-SMan. Vice versa, if at least one of the two constraints is not satisfied and streams can be modeled as elastic $S_i = (C_i, D_i \in [D_i^{min}, D_i^{max}], \varepsilon_i)$, then El-SMan can be used to compute for each S_i the value of $n_i \in [n_i^{min}, n_i^{max}]$ such that both constraints are met. For a given delay bound, the value of n_i can be derived from the formulation of $D_{n_i,max}$ provided in Koubaa et al.

[2007] through the following equation:

$$n_i^2 T_s + n_i \left(D_{n_i,max} - BI \right) - \frac{C_i}{\lambda DC - w_{idle}} = 0, \tag{21}$$

where w_{idle} can be computed as $\frac{T_S - T_a}{T_S}$ and $\lambda = 1/16$. The values of n_i^{min} (U_i^{min}) and n_i^{max} (U_i^{max}) can be derived from Equation (21) by imposing $D_{n_i,max} = D_i^{min}$ and $D_{n_i,max} = D_i^{max}$, respectively.

It is worth observing that El-SMan is not an alternative solution to the energy-saving mechanism used in IEEE 802.15.4-compliant protocols. Instead, it is an algorithm that can be used in conjunction with the energy-saving mechanism provided by the protocol to compute the protocol parameters with the purpose of satisfying both timing and energy constraints. In other words, El-SMan does not affect the underlying energy-saving mechanism, but it acts on the transmission bandwidth allocated to each node to balance the energy consumption and communication performance. This claim holds for any protocol, including RI-EDF and WBuST.

4.3. Dynamic Adaptation of El-SMan

This subsection discusses the advantages of El-SMan with respect to the simplex algorithm [Murty 1983], which is a well-known method providing an optimal solution to the linear programming problem defined by Equation (9). The main point in favor of El-SMan is its polynomial-time complexity in the worst case, in contrast with the worst-case exponential time complexity of the simplex method. This is particularly important for dynamic environments, such as wireless networks, where the network's nodes can dynamically create and remove message streams. In this case, the cluster stream sets can change their configuration over time, requiring a recomputation of the stream utilizations at every configuration change. Another situation in which stream sets can change their configuration is when a node moves from a cluster to another. In this case, a stream is subtracted from the set of the starting cluster and added to the set of the destination cluster. A further reason to dynamically compute the stream utilizations comes from the fact that the energy model, as proposed in this work, can be quite conservative in the power consumption estimation. In this situation, starting from the current available energy and the current time interval needed to reach the desired lifetime, the nodes can periodically recompute the stream utilizations to better exploit the energy management mechanism. The demand of dynamic adaptation, due to the reasons highlighted previously, requires the use of a low time complexity algorithm that can be executed online by, for instance, each cluster coordinator. This makes El-SMan more suitable for the dynamic environment than the simplex method.

It is worth pointing out that, in order to compute the stream configurations, each coordinator needs the knowledge of the stream parameters of nodes belonging to its cluster and the parameters of the streams generated to communicate with the coordinators/routers of adjacent clusters. Such information, however, can be sent during the network initialization phase, so that only variations need to be sent at runtime whenever a node wants to join or leave the cluster. A further consideration concerns the time needed by the coordinator to communicate the new stream configurations to the nodes. This can be done by broadcast messages transmitted by the coordinator or, in case of beacon/polling-based networks, by adding this information to beacon/polling messages sent by the coordinator.

5. ALGORITHM PROPERTIES

This section reports some observations and results on the El-SMan properties. First, it is worth observing that in general, the proposed algorithm (El-SMan) provides a solution for a more relaxed problem than that defined at the end of Section 3 (refer to the optimization problem in Equation (9)). In other words, El-SMan provides a utilization vector $\overrightarrow{U} = (U_1, U_2, \ldots, U_n)$ such that:

—$U = \sum_{i=1}^{n} U_i \leq U^*$, that is, the message deadlines are met;
—$\forall i$, $U_i^{min} \leq U_i \leq U_i^{max}$;
—$L_{net} = L_{net}^d$, that is, the system lifetime is met; and
—$U \leq U^{opt}$,

where U^{opt} is the solution of the optimization problem (Equation (9)). The algorithm provides a solution such that the stream set Γ is feasible in the sense of Definition 3.1, but this solution might not be the optimal one.

Observe that, if the system lifetime is met when for every i, $U_i = U_i^{min}$, El-SMan certainly produces a solution because, in the worst case, this is the solution provided by the algorithm. Conversely, if the system lifetime is not met with the minimal stream utilizations, no solution exists.

5.1. Properties of the Algorithm Parameters

For the sake of clarity, the remaining part of the article considers homogenous nodes (nodes with the same radio transceiver), that is, $P_i^{rs} = P^{rs}$, $P_i^{ts} = P^{ts}$ for all i. Moreover, it assumes $\Delta_i \simeq \Delta$ for all i.

Let $P_{tot}^{max} = \sum_{i=1}^{n} \frac{E_i^0}{L_{net}^d}$ be the maximum power that the system can waste while guaranteeing the desired lifetime. The following result provides an optimal solution to the problem stated by Equation (9).

THEOREM 5.1. *Given the optimization problem defined in Equation (9), for $i = (1, \ldots, n)$, if the following assignment:*

$$U_i = \frac{1}{n} \cdot \frac{P_{tot}^{max} - n\Delta}{P^{ts} + (n-1)P^{rs}} \tag{22}$$

satisfies the problem constraints, then it represents a solution of the optimization problem (Equation (9)). That is, it maximizes the network utilization guaranteeing the desired lifetime.

PROOF. See appendix. □

The following properties are helpful to improve the algorithm performance, especially when it is used in embedded systems where the computation power is limited.

The next lemma is used to prove the subsequent theorem.

LEMMA 5.2. *Given a message stream set $\Gamma = \{S_1, S_2, \ldots, S_n\}$, for any (i, v) with $i \neq v$, if $P^{ts} \geq P^{rs}$, then $P_i \geq P_v$ if and only if $U_i \geq U_v$. If $P^{ts} \leq P^{rs}$, then $P_i \geq P_v$ if and only if $U_i \leq U_v$.*

PROOF. See appendix. □

The following theorem provides a way to select the coefficients ϵ_i, in particular the elastic coefficient ϵ_y of stream S_y, so that the algorithm converges by one iteration; that is, the cycle (from line 9 to line 19 in Figure 4) is executed just once.

THEOREM 5.3. *Given an elastic stream set $\Gamma = \{S_1, S_2, \ldots, S_n\}$, let S_y be the stream with the shortest lifetime and let $k_i = \frac{U_y - U_i}{U^0 - U^d}$. If $P^{ts} \geq P^{rs}$, then El-SMan provides a* *solution by just one iteration if and only if for all i:*

$$\epsilon_y \leq \epsilon_i \frac{1+k_i}{1-k_i} + \frac{k_i}{1-k_i}\varepsilon^{i,y}. \tag{23}$$

If $P^{ts} \leq P^{rs}$, El-SMan provides a solution by just one iteration if and only if for all i:

$$\epsilon_y \geq \epsilon_i \frac{1+k_i}{1-k_i} + \frac{k_i}{1-k_i}\varepsilon^{i,y}, \tag{24}$$

where $\varepsilon^{i,y} = (\sum_{j=1;j\neq i,y}^{n} \epsilon_j) = (\sum_{j=1}^{n} \epsilon_j) - \epsilon_i - \epsilon_y$.

PROOF. See appendix. □

It is worth noticing that, from the inequality in Equation (23), if for some i $k_i > 1$, $P^{ts} \geq P^{rs}$, then $\epsilon_y < 0$. This means that there are no solutions because ϵ_y must be nonnegative. Further, from the inequality in Equation (24), if $k_i = 1$ and $P^{ts} \leq P^{rs}$, there are no solutions because $\epsilon_y \geq \infty$.

The following corollary guarantees that an available solution for ϵ_y always exists; in other words, it guarantees that the bad conditions described previously are impossible.

COROLLARY 5.4. *Given an elastic stream set $\Gamma = \{S_1, S_2, \ldots, S_n\}$, let S_y be the stream with the shortest lifetime, defined $k_i = \frac{U_y^0 - U_i^0}{U^0 - U^d}$; if $P^{ts} \geq P^{rs}$, then for any i, $0 \leq k_i \leq 1$. If $P^{ts} \leq P^{rs}$, then for any i, $k_i \leq 0$.*

PROOF. See appendix. □

From Theorem 5.3, in order to get the coefficients ϵ_i, it is necessary to solve a linear inequalities system. However, as the function $f_\epsilon(k_i) = \epsilon_i \frac{1+k_i}{1-k_i} + \frac{k_i}{1-k_i}\varepsilon^{i,y}$ is increasing monotonic for $0 \leq k_i < 1$ and decreasing monotonic for $k_i < 0$, once it chooses each ϵ_i except ϵ_y, defined $k_{min} = \min_i(k_i)$ and $k_{max} = \max_i(k_i)$, it is sufficient to select ϵ_y as indicated by the inequalities in Equations (23) and (24), solving just one inequality, considering only k_{min} and k_{max}, respectively. In this way, the relations of Theorem 5.3 can be satisfied without solving a linear inequalities system.

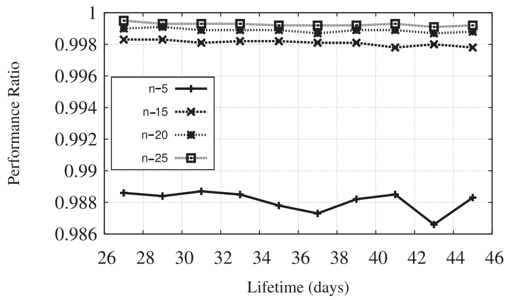
6. SIMULATION RESULTS

This section presents a set of simulation experiments carried out to test the performance of the El-SMan algorithm under different conditions, varying the desired system lifetime (measured in days) and the number of nodes involved in the network. P^{tx} and P^{rx} have been derived by multiplying the operating voltage by the currents sunk in the various working modes. These values have been obtained from the data sheet of the *CC*2420 [2016], which is a widely used radio transceiver.

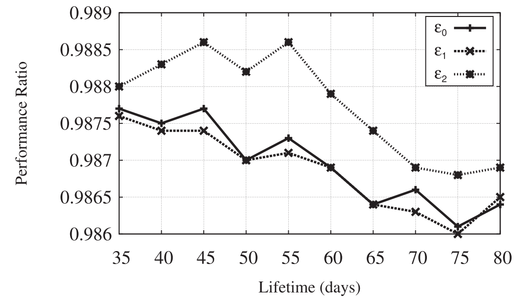
Considering that the network nodes are grouped into clusters, it is worth pointing out that the energy consumption of a node only depends on the message streams of the nodes belonging to same cluster, including the coordinator. The energy consumption of the coordinator must also consider those messages sent to and received from the coordinators of adjacent clusters. It turns out that El-SMan is executed in each network cluster taking into account the message streams handled by the nodes in that cluster. Therefore, the simulation experiments have been carried out considering a set of n message streams representing the streams associated with the nodes of a single cluster.

The algorithm has been evaluated calculating the performance ratio p_{ratio} between the total utilization U^d, provided by El-SMan, and U^{opt}, provided by the simplex algorithm:

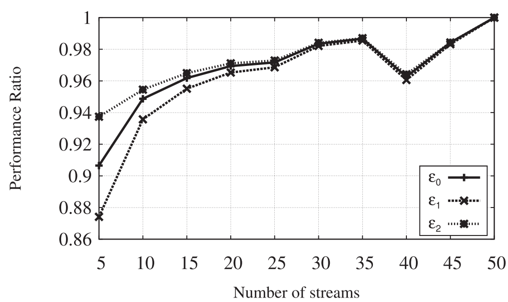
$$p_{ratio} = \frac{U^d}{U^{opt}}. \tag{25}$$



(a) Performance ratio versus lifetime and number of streams.



(b) Performance ratio versus lifetime and elastic coefficients selection.



(c) Performance ratio versus number of streams.

Fig. 7. Simulation results.

Each simulation was performed generating, for each stream S_i, the utilizations U_i^{max} and U_i^{min} using the UUniFast algorithm proposed by Bini and Buttazzo [2004]. The maximum total utilization, that is, the sum of U_i^{max}, was $U^* = 1$ for each simulation run.

Usually, in the elastic model, the coefficients ϵ_i represent the inverse of the relative importance of each stream. For instance, some traffic streams can be compressed a little because they are important for the system operation; some others could be less important and hence can be compressed more. In general, the importance of a stream depends on the application. In these simulation experiments, the elastic coefficients ϵ_i have been generated in three different ways:

(1) $\epsilon_i = \textit{random number}$, uniformly distributed in [0, 1];
(2) $\epsilon_i = \frac{U_i^{max} - U_i^{min}}{\max_i(U_i^{max} - U_i^{min})}$; and
(3) $\epsilon_i = \frac{P_i^{max} - P_i^{min}}{\max_i(P_i^{max} - P_i^{min})}$;

The second assignment selects the coefficients based on the utilization range of each stream S_i normalized by the maximum utilization range: the higher the range is, the higher the elastic coefficient. Similarly, the last method selects the coefficients based on the normalized power range of each stream.

The average performance ratio obtained in the experiments described in the following has been computed over 500 simulation runs, for each value of lifetime and number of nodes.

Figure 7(a) shows the average performance ratio obtained by varying the desired lifetime. Each curve represents p_{ratio} for a specific number of streams. As can observed, the performance ratio does not vary significantly as the desired lifetime increases. For

Table I. Current Consumptions

Working Mode	Current
Transmission	$17.1mA$
Receiving	$18.8mA$
Sleep	$1.8mA$

$n = 15, 20, 25$, p_{ratio} is greater than 0.9975. It means that the total utilization provided by El-SMan is, on average, 0.25% smaller than the optimal one provided by the simplex algorithm. For $n = 5$, the performance ratio is slightly lower than in the other cases. The reason is that the higher the number of nodes, the higher the probability is that the algorithm provides a solution similar to that of Theorem 5.1, which is an optimal one.

Figure 7(b) represents the average performance ratio obtained by varying the desired lifetime, where each curve represents the values obtained be generating the elastic coefficients as described earlier. In this experiment, the number of streams is fixed and equal to 5. It can be noted that the best performance is achieved when the elastic coefficients depend on the power range of the associated stream: the higher the power range is, the higher the stream elasticity, and hence, the more the stream is compressed. Notice that, also in this experiment, p_{ratio} is greater than 0.986.

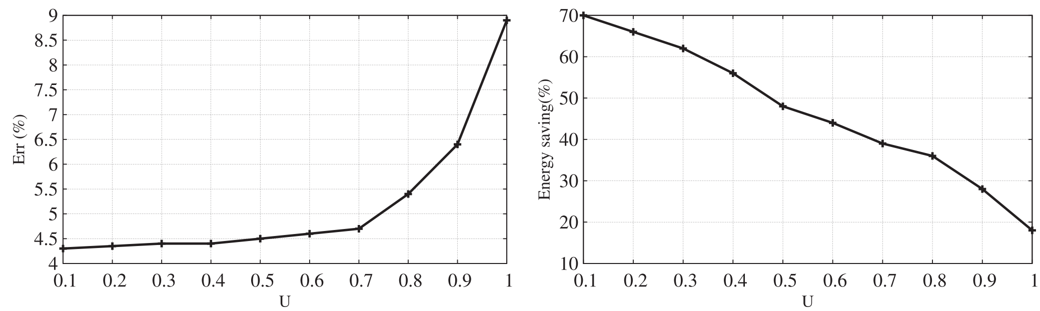
The average performance ratio obtained varying the number of streams is shown in Figure 7(c). The three curves are obtained with the three ways used to generate the coefficients ϵ_i. Observe that the curves differ a little when the number of nodes ranges from five to 20. After that, the curves tend to overlap for the same reason explained earlier: as the number of nodes increases, the algorithm tends to provide the optimal solution defined by Theorem 5.1, regardless of the method used to generate the elastic coefficients.

7. EXPERIMENTAL RESULTS

A set of experiments was performed on a real set of nodes to test the feasibility of the proposed method in managing and predicting the energy consumption. The testbed is composed by 10 FLEX boards [2016] equipped with a 16-bit microcontroller and the CC2420 radio module, which is an IEEE 802.15.4-compliant transceiver. The communication stack was realized by implementing the RI-EDF protocol. Since El-SMan works on the stream set of each network cell, it is not necessary to create a multicell network to test the algorithm; thus, the 10 boards were grouped into a single cell. The firmware has been written in C under the real-time kernel ERIKA Enterprise [2016]. In this implementation of RI-EDF [Crenshaw et al. 2005], the time is divided into slots, representing the time unit at the network level. In particular, the slot dimension is equal to $4ms$. This value depends on the specific hardware used in this experimentation, mainly the radio transceiver. In this setting, the slot dimension is sufficient to send a packet of 125 bytes including the header. The nodes transmit their traffic in broadcast without acknowledgment packets.

The power consumption of a select node was measured by a current-sensing circuit connected to its radio module. The measurement circuit acquires the average current sunk by the radio. The average power is computed by multiplying the current by the operating voltage. In the same way, the electric current consumed by the radio in transmission, receiving, and sleep modes was measured and is reported in Table I. These values, slightly different from those provided by the transceiver data sheet, are used to compute P_i^{tx}, P_i^{rx}, and P_i^{sl} in the energy model.

All experimental results are obtained by varying the total stream utilization U from 0.1 to 1 with a step of 0.1. A message stream was assigned to each node, for a



(a) Energy model error.

(b) Energy saving.

Fig. 8. Experimental results.

total amount of 10 streams. For each value of U, 10 tests were performed monitoring the average current consumption over an interval of 20 minutes. A new task set was generated for each test in the following way. The stream utilizations were randomly generated within a uniform distribution. For each U_i, a relative deadline D_i was randomly selected in the interval [20 slots; 480 slots] with a step of five slots. T_i is assumed equal to D_i and the message length M_i is computed multiplying U_i by T_i and rounding the result to the nearest integer.

A first set of experiments was carried out to verify the ability of the energy model, provided by Equation (2), on estimating the energy consumption. In general, it is not an easy task to derive U_i^{ctrl} and U_i^{sw} without complete knowledge of the channel access schedule; thus, the experiments considered a simplified version of the model where $\Delta_i = P_i^{sl}$. The validity of the model was verified by measuring the average power consumption of a selected node i, which was compared with that computed by the model. Let $\bar{P}_i$ be the average power consumption measured during the test; the model error $Err(\%)$ has been computed as

$$Err(\%) = 100\left(1 - \frac{P_i}{\bar{P}_i}\right). \tag{26}$$

As said previously, for each value of U, 10 tests were performed to obtain $Err(\%)$ as the maximum value among all tests. Figure 8(a) shows the results. Observe that, as long as the stream set utilization is not greater 0.7, the maximum model error is less than 5%. For greater values of U, the error increases up to 9%. This means that in the worst case, the network lifetime estimated by the model is between 5% and 9% greater than the real one depending on the network load. The difference between the predicted energy consumption and the actual energy consumption is mainly due to the choice of $\Delta_i = P_i^{sl}$. In this way, the model does not consider the energy wasted by the transceiver to switch between working modes; that is, it assumes $U_i^{ctrl} = U_i^{sw} = U^{sw} = 0$. It turns out that, when such parameters can be computed, the model error can be significantly reduced.

Figure 8(b) shows the percentage of energy saving of the selected node when it exploits idle intervals to save energy through the sleep mode, with respect to the case when the sleep mode is not utilized. It can be noted that the percentage of saving depends on the stream set utilization: the lower the utilization is, the higher the idle time, that is, the higher the sleep time. In particular, for $U = 0.1$, the node can save 70% of the energy and, as U increases, the percentage of energy saving decreases down to 20% when the channel is fully loaded, that is, $U = 1$. Note that, although for $U = 1$ there is no idle time, if a slot is partially used, a node can go on sleep in the unused portion of the slot. A slot is partially used whenever a node transmits a packet shorter than the maximum length of 125 bytes.

8. CONCLUSION AND FUTURE WORK

This article analyzed the problem of saving energy at the communication level for a distributed embedded system composed by a network of wireless nodes. It assumed that each node is associated with a message stream, described through an elastic stream model, in which each stream utilization (bandwidth) can vary in a predefined interval. Given the available energy on each node, an energy consumption model that allows estimating the network lifetime is derived. This model takes into account the energy wasted to deliver messages by the radio transceiver. This energy is managed at the MAC level.

After having defined the system lifetime, the article introduced an algorithm (El-SMan) that selects the bandwidth of each stream within a specified range, so that both real-time constraints and the network lifetime can be guaranteed. It is also shown that, if the elastic coefficients of the streams are appropriately chosen, the algorithm ends by just one iteration.

El-SMan was tested by simulation in comparison with the simplex algorithm, which is an optimal but more complex method. This comparison has shown that El-SMan provides suboptimal solutions, but the difference between the solutions provided by the two algorithms is not greater than 0.25%. Another advantage of El-SMan with respect to the simplex method regards its lower worst-case execution time, which makes El-SMan suitable for dynamic environments, such as those represented by wireless networks.

The proposed energy model and El-SMan are directly applicable to TDMA scheduling approaches, such as the RI-EDF protocol. The major differences among various scheduling-based protocols can reside in the computation of U^{sw} and U^{ctrl}, which strictly depend on the MAC protocol under consideration. The energy model was also validated through experiments with a set of embedded boards. The experiments show that the estimation on the energy consumption differs from the actual value for no more than 9%; it depends on the stream set utilization. If U is not greater than 0.9, the error is less than 6.5%.

As a future work, we plan to extend this work in several directions. First, as pointed out in Section 3.3, El-SMan assumes that a node i sets the transmission power P_i^{tx} to the minimum level that guarantees the network connectivity. This could be done by tracking the $RSSI$ values of received messages and by setting the power to a level that guarantees $RSSI$ values above a minimum threshold. In this way, each node i could set P_i^{tx} dynamically in accordance with link conditions and energy-saving requirements. Another possible improvement is to derive a method for integrating El-SMan with energy-saving mechanisms that can be used at higher levels of the node architecture. Examples are the energy-aware scheduling algorithms proposed in the literature to allocate the CPU time to node tasks.

APPENDIX

PROOF OF THEOREM 5.1.

PROOF. From the energy model, for all $i = 1, \dots, n$ and $j = 1, \dots, n$ with $j \neq i$, it is possible to define U_i as a function of P_i and U_j:

$$U_i = \frac{P_i - \sum_{j \neq i}^{n} U_j P^{rs} - \Delta}{P^{ts}}.$$

Summing up for all i both sides of the equation, and remembering that $U = \sum_i U_i$, it turns out that

$$U = \frac{\sum_{i=1}^{n} P_i - (n-1)UP^{rs} - n\Delta}{P^{ts}}.$$

Then, defining $P_{tot} = \sum_{i=1}^{n} P_i$ and gathering the terms:

$$U = \frac{P_{tot} - n\Delta}{P^{ts} + (n-1)P^{rs}}. \tag{27}$$

Replacing P_{tot} with P_{tot}^{max} in the previous equation, it is possible to derive the maximum channel utilization that the system can guarantee, given the desired lifetime L_{net}^{d} and the energy available for each node i (E_i^0). Hence, it turns out that if the assignment given by Equation (22) satisfies the problem constraints, then such an assignment provides an optimal solution because, in this case, the total channel utilization is provided by Equation (27). □

Proof of Lemma 5.2.

Proof. From the energy model, $P_i \geq P_v$ means that

$$U_i P^{ts} + \sum_{j=1, j\neq i}^{n} U_j P^{rs} + \Delta \geq U_v P^{ts} + \sum_{j=1, j\neq v}^{n} U_j P^{rs} + \Delta,$$

and then

$$(U_i - U_v)P^{ts} + \left(\sum_{j=1, j\neq i}^{n} U_j P^{rs} - \sum_{j=1, j\neq v}^{n} U_j P^{rs} \right) \geq 0$$

$$(U_i - U_v)P^{ts} + (U - U_i - U + U_v)P^{rs} \geq 0$$

$$(U_i - U_v)P^{ts} \geq (U_i - U_v)P^{rs}. \tag{28}$$

From the inequality in Equation (28), if $P^{ts} \geq P^{rs}$, then $P_i \geq P_v$ if and only if $U_i \geq U_v$. If $P^{ts} \leq P^{rs}$, then $P_i \geq P_v$ if and only if $U_i \leq U_v$. □

Proof of Theorem 5.3

Proof. The inequality in Equation (23) is proved first.

(If) Suppose that the algorithm converges by one iteration; that is, after the first iteration, $P_y^d = E_i^0 / L_{net}^d$ and $\forall i$ $P_y^d \geq P_i$. From Lemma 5.2, $\forall i$ if $P^{ts} \geq P^{rs}$, then $P_y^d \geq P_i$ if and only if $U_y \geq U_i$, that is, $U_y - U_i \geq 0$. From Equation (10),

$$U_y - U_i = U_y^0 - U_i^0 - \frac{U^0 - U^d}{\varepsilon}(\epsilon_y - \epsilon_i) \geq 0$$

$$U_y^0 - U_i^0 \geq \frac{U^0 - U^d}{\varepsilon}(\epsilon_y - \epsilon_i)$$

$$\frac{U_y^0 - U_i^0}{U^0 - U^d}\varepsilon \geq (\epsilon_y - \epsilon_i).$$

Since $\varepsilon = \sum_{j=1}^{n} \epsilon_j = \varepsilon^{i,y} + \epsilon_i + \epsilon_y$ and $k_i = \frac{U_y^0 - U_i^0}{U^0 - U^d}$, then if the algorithm ends by one iteration:

$$(\epsilon_y - \epsilon_i) \le k_i \left(\left(\sum_{j \neq i,y} \epsilon_j \right) + \epsilon_i + \epsilon_y \right)$$
$$\epsilon_y (1 - k_i) \le \epsilon_i (1 + k_i) + k_i \varepsilon^{i,y}$$
$$\epsilon_y \le \epsilon_i \frac{1 + k_i}{1 - k_i} + \frac{k_i}{1 - k_i} \varepsilon^{i,y}.$$

(Only if) Suppose that the algorithm ends by one iteration and for some i, $\epsilon_y > \epsilon_i \frac{1+k_i}{1-k_i} + \frac{k_i}{1-k_i} \varepsilon^{i,y}$, and then, following the inverse process, it finds out that in this case, $U_y < U_i$, that is, $P_y^d = E_i^0 / L_{net}^d < P_i$, and hence the algorithm has not terminated: this is a contradiction, and therefore, the thesis follows.

Now, suppose that the algorithm converges by one iteration, and hence, $P_y^d = E_i^0 / L_{net}^d$ and $\forall i$ $P_y^d \ge P_i$. To prove the inequality in Equation (24), it is sufficient to observe that, from Lemma 5.2, if $P^{ts} \le P^{rs}$, then $P_y^d \ge P_i$ if and only if $U_y \le U_i$, that is, $U_y - U_i \le 0$. The rest of the proff is similar to that for the inequality in Equation (23), where it is sufficient to exchange the signs ($\le$) and ($\ge$). □

Proof of Corollary 5.4.

Proof. Let's start defining a function $f_\epsilon(k_i) = \epsilon_i \frac{1+k_i}{1-k_i} + \frac{k_i}{1-k_i} \varepsilon^{i,y}$, where $i \neq y$ and ϵ_i, $\varepsilon^{i,y}$ are given parameters.

In the first case, where $P^{ts} \ge P^{rs}$, from Lemma 5.2, at each step the algorithm chooses y so that for all i, $U_y^0 \ge U_i^0$, and then $k_i \ge 0$. The following can be observed:

—From Theorem 5.3, if $k_i < 1$, then to converge by just one iteration, it needs that $\epsilon_y \le f_\epsilon(k_i) < \infty$. For $k_i = 1$, $f_\epsilon(k_i) = \infty$. Thus, it can be chosen as $\epsilon_y \le f_\epsilon(k_i) \le \infty$ for $0 \le k_i \le 1$.

—If $k_i > 1$, from the inequality in Equation (23), to converge by just one iteration, it needs that $\epsilon_y \le f_\epsilon(k_i) < 0$. Note that, ϵ_y being an elastic coefficient, $\epsilon_y < 0$ has no meaning. However, it is not difficult to see that if for all i, $0 \le \epsilon_y \le \epsilon_i$ and then $U_y \ge U_i$; that is, the algorithm finishes by just one iteration. Therefore, the inequality in Equation (23) being a necessary and sufficient condition, it follows that either $0 \le k_i \le 1$ or the inequality in Equation (23) is incorrect. Since the correctness of this latter inequality has been proved, the first part of the thesis follows.

In the second case, $P^{ts} \le P^{rs}$. From Lemma 5.2, at each step the algorithm chooses y so that for all i, $U_y^0 \le U_i^0$, and this means that $k_i \le 0$. □

REFERENCES

CC2420. 2016. http://www.ti.com.

FLEX Boards. 2016. http://www.evidence.eu.com/flex-daughter-boards.html.

muRata DR3300. 2016. http://wireless.murato.com.

RTOS Erika Enterprise. 2016. http://erika.tuxfamily.org.

M. Adamou, I. Lee, and I. Shin. 2001. An energy efficient real-time medium access control protocol for wirelss ad-hoc networks. In *Proceedings of the 22th IEEE Real-Time Systems Symposium (RTSS'01)*.

G. Anastasi, M. Conti, M. Di Francesco, and V. Neri. 2010. Reliability and energy efficiency in multi-hop IEEE 802.15.4/ZigBee wireless sensor networks. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'10)*.

E. Bini and G. Buttazzo. 2004. Biasing effects in schedulability measures. In *Proceedings of the 16th Euromicro Conference on Real-Time Systems (ECRTS'04)*.

D. M. Blough and P. Santi. 2002. Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks. In *Proceedings of MOBICOM 2002*.

G. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni. 2002. Elastic scheduling for flexible workload management. *IEEE Trans. Comput.* 51, 3 (March 2002), 289–302.

M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo. 2002. An implicit prioritized access protocol for wireless sensor networks. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS'02)*.

J.-J. Chen and T.-W. Kuo. 2007. Procrastination determination for periodic real-time tasks in leakage-aware dynamic voltage scaling systems. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD'07)*. 289–294.

Y. Chen and Q. Zhao. 2005. On the lifetime of wireless sensor networks. *IEEE Commun. Lett.* 9, 11 (November 2005), 976–978.

T. L. Crenshaw, A. Tirumala, S. Hoke, and M. Caccamo. 2005. A robust implicit access protocol for real-time wireless collaboration. In *Proceedings of the IEEE Euromicro Conference on Real-Time Systems (ECRTS'05)*.

D. D. Demarch and L. B. Becker. 2007. An integrated scheduling and retransmission proposal for firm real-time traffic in IEEE 802.11e. In *Proceedings of the IEEE Euromicro Conference on Real-Time Systems (ECRTS'07)*.

M. Di Francesco, G. Anastasi, M. Conti, S. K. Das, and V. Neri. 2011. Reliability and energy-efficiency in IEEE 802.15.4/ZigBee sensor networks: An adaptive and cross-layer approach. *IEEE J. Selected Areas Commun.* 29, 8 (September 2011), 1508–1524.

G. Franchino, G. Buttazzo, and M. Marinoni. 2010. An energy-aware algorithm for TDMA MAC protocols in real-time wireless networks. In *Proceedings of the 5th IEEE International Symposium on Embedded Systems (SIES'10)*.

G. Franchino and G. C. Buttazzo. 2012. WBuST: A real-time energy-aware MAC layer protocol for wireless embedded systems. In *Proceedings of the 17th IEEE International Conference on Emerging Technology and Factory Automation (ETFA'12)*.

Tian He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. 2003. SPEED: A stateless protocol for real-time communication in sensor networks. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS'03)*.

T. D. Hoa and D.-S. Kim. 2012. Minimum latency and energy efficiency routing with lossy link awareness in wireless sensor networks. In *Proceedings of the 9th IEEE International Workshop on Factory Communication Systems (WFCS'12)*. 2006. *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. Technical Report. IEEE-SA Standards Board.

A. Koubaa, M. Alves, and E. Tovar. 2006. Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. In *Proceedings of the 27th IEEE Real-Time Systems Symposium (RTSS'06)*.

A. Koubaa, M. Alves, and E. Tovar. 2007. Energy/delay trade-off of the GTS allocation mechanism in IEEE 802.15.4 for wireless sensor network. *Int. J. Commun. Syst.* 20, 7 (July 2007), 791–808.

C. L. Liu and J. W. Layland. 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. *JACM* 20, 1 (February 1973), 46–61.

J. Ma, S. H. Kim, and D. Kim. 2012. Tame: Time window scheduling of wireless access points for maximum energy efficiency and high throughput. In *Proceedings of the 9th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCAS'12)*.

K. G. Murty. 1983. *Linear Programming*. Wiley.

A. Rowe, R. Mangharam, and R. Rajkumar. 2006. RT-Link: A time-synchronized link protocol for energy-constrained multi-hop wireless networks. In *Proceedings of the 3rd IEEE International Conference on Sensors, Mesh and Ad Hoc Communications and Networks (IEEE SECON'06)*.

M. M. Sobral and L. B. Becker. 2008. A wireless hybrid contention/TDMA based MAC for real-time mobile application. In *Proceedings of the ACM Symposium on Applied Computing (SAC'08)*.

J. A. Stankovic, T. He, T. F. Abdelzaher, M. Marley, G. Taoand, S. Son, and C. Lu. 2001. Feedback control scheduling in distributed systems. In *Proceedings of the 22th IEEE Real-Time Systems Symposium (RTSS'01)*.

E. Toscano and L. Lo Bello. 2012. Comparative assessments of IEEE 802.15.4/ZigBee and 6LoWPAN for low-power industrial WSNs in realistic scenarios. In *Proceedings of the 9th IEEE Symposium on Computers and Communications (ISCC'10)*.

R. Xu, D. Mosse, and R. Melhem. 2007. Minimizing expected energy consumption in real-time systems through dynamic voltage scaling. *ACM Trans. Comput. Syst.* 25, 4 (2007), 9:1–9:40.

W. Ye, J. Heidemann, and D. Estrin. 2004. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.* 12, 3 (June 2004), 493–506.

B. Zhao and H. Aydin. 2009. Minimizing expected energy consumption through optimal integration of DVS and DPM. In *Proceedings of the 2009 International Conference on Computer-Aided Design (ICCAD'09)*. ACM, New York, New York, 449–456.

Received July 2014; revised November 2015; accepted November 2015