

An Energy-Aware Algorithm for TDMA MAC Protocols in Real-Time Wireless Networks

Gianluca Franchino, Giorgio Buttazzo and Mauro Marinoni

firstname.lastname@sssup.it

Scuola Superiore S. Anna Pisa, Italy

Abstract—In distributed embedded systems operated by battery, energy management is a critical issue that has to be addressed at different architecture levels. For systems that tightly interact with the environment, an additional goal is to enforce a set of real-time constraints to guarantee a desired performance. A lot of research has focused on power management at the communication level, especially for MAC protocols. However, not many authors considered both real-time and energy requirements in wireless communication systems. In this paper we present EL-MAC, an elastic energy-aware algorithm at the MAC level for wireless distributed systems with real-time constraints. Under this framework, each node can adapt its bandwidth requirements to balance performance versus energy consumption, taking both lifetime and message deadlines into account. We describe the algorithm for a generic TDMA MAC protocol.

I. INTRODUCTION

Embedded systems operated by battery are exponentially increasing, including mobile phones, multimedia players, portable devices, sensor networks, teams of mobile robots, and distributed communication systems. Some of these systems are required to interact with the environment, reading sensors, reacting to events, and performing proper control actions. Consider, for example, a distributed monitoring system consisting of tens or hundreds of mobile nodes delivered in a specific area to detect intrusions, or abnormal situations. To achieve this goal, each node has to be 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 communication devices (to coordinate with the other nodes and transmit the information to the base station). From one hand, the interaction with the environment creates implicit timing constraints on the application that have to be enforced on the node activities in order to achieve a desired performance. From the other hand, since nodes are powered by batteries, the available energy must be carefully managed to prolong the system lifetime long enough to complete the mission.

Unfortunately, real-time and energy requirements have contrasting objectives and they cannot be considered separately during system design. For instance, meeting real-time constraints on sensory and control activities would require the onboard processor to use a suitable scheduling algorithm and to run fast enough in order 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 would require the nodes to have a large transmission range to guarantee a single hop connection,

even though a large transmission range consumes more power, reducing the system autonomy.

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.

To minimize energy consumption, power should be saved at different architecture levels. At the application level, specific devices can be turned off, or configured at a proper operating low-power mode (if any), when they are not used for a sufficiently long interval of time. At the operating system level, suitable scheduling and resource management algorithms can be adopted to execute tasks at the minimum speed that guarantees the required performance constraints. 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.

In the market there exist several transceivers single-chip suitable to build small radio devices. These chips have some smart features that can be exploited to design energy-aware transmission protocols [1], [2]. For instance several devices provide:

- RSSI (Receiving Signal Strength Indicator), which is a value proportional to the strength of the received RF signal. It can give a greedy esteem 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: sleep, receiving, and transmitting.

In this paper, we describe a methodology to save energy at the communication level, precisely at the Medium Access Control (MAC) level, in wireless networks with real-time requirements. As proposed in [3], we take into account the following main sources of energy waste:

- *Collision*. If a packet is corrupted it has to be resent, 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.

In some application, idle listening can be the main cause of energy waste, since nodes can stay for a long time without sending messages, as the case of a sensor network where no events are generated in the environment.

Beside the energy waste sources described above, in this work we also consider 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 viceversa, could be greater than that needed to stay always active.

A. Contributions and summary

In this work, we propose an energy model and a related algorithm to reduce the energy consumption in a wireless network, with the objective of reaching a given system lifetime while guaranteeing periodic messages deadlines. In particular, we present EI-MAC, an elastic energy-aware algorithm at the MAC level, where each node can adapt its bandwidth requirements to balance performance versus energy consumption. The proposed methodology is designed for Time Division Multiplexing Access (TDMA) protocols, and for a network traffic generated by real-time streams.

The remainder of the paper is organized as follows. Section II describes the related work. Section III introduces the problem formulation, and describes both the power consumption and the message stream models. Section IV illustrates the EI-MAC algorithm and presents an applicative example. Section V analyzes the properties of the algorithm. Section VI presents some simulation results. Finally, Section VII concludes the paper and presents some future developments.

II. RELATED WORK

The issues of real-time communication and energy saving over wireless networks have received much attention during the last years. However, not many authors addressed both problems simultaneously. Adamou et al. [4] proposed a MAC protocol (PARMAC) that provides a guaranteed bandwidth for real-time traffic, while reducing the energy consumption. Time is divided in 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 less 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,

nor compare the energy performance with respect to other protocols.

Caccamo et al. [5] proposed a cellular network architecture with a MAC protocol based on the Earliest Deadline First algorithm (EDF) [6]. 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 EDF 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. [7] 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 which 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.

Unlike the previous protocols which are mainly focused on the MAC layer, SPEED [8] 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 delivering, through a feedback control scheduling mechanism [9]. Energy savings performance is analyzed comparing the SPEED routing algorithm with other routing solutions available in the literature.

Koubaa et al. [10] analyzed the power efficiency and the timeliness of the IEEE 802.15.4 standard under the GTSs mechanism. They also proposed a method to select the protocol parameters, to trade the power efficiency and the delay bound guarantees provided by this technology.

A further interesting work is the RT-Link protocol, proposed by Rowe et al. [11], 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 [12] and [13].

III. PROBLEM AND MODELS

In this work we consider a real-time distributed system composed by n computational nodes sharing a communication channel, whose access is managed through a MAC layer protocol, using a TDMA scheduling algorithm.

A. Network traffic model

The traffic generated by each node i is modeled by a message stream which 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 amount of time required to transmit a message in the stream.
- T_i is the inter-arrival 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 j -th message in stream S_i will be generated at time $t_{i,j} = t_{i,1} + (j-1)T_i$, where $j = 1, 2, \dots$
- 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.

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.

For the sake of simplicity, we assume 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, we consider the communication system composed by a stream set $\Gamma = \{S_1, S_2, \dots, S_n\}$, where each stream can be a real-time or a bandwidth-guaranteed stream.

The total channel utilization, U , which measures the total demand placed on the system by the stream message set is

$$U = \sum_{i=1}^n U_i \quad (1)$$

where $U_i = U_i^{RT} = C_i/T_i$ in case of a real-time stream, or $U_i = U_i^{BG}$ in case of a bandwidth guaranteed streams.

It is reasonable for several applications to assume that the stream parameters are constant for each message, but there are situations in which periodic messages could be delivered at different rates in different working conditions. 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 run time, whenever a new stream cannot be accepted in the system because its utilization would make the channel occupation greater than 1, instead of rejecting the stream, the system could try to reduce the other streams utilizations, decreasing the channel occupation in order to accept the new request. The same reasoning can be applied for bandwidth-guaranteed streams.

Elastic scheduling was first proposed in [14] for real-time tasks. We extend this framework 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 paper we assume 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. Therefore, an elastic stream is denoted as: $\hat{S}_i = (U_i^{min}, U_i^{max}, \epsilon_i)$.

B. Energy model

From now on, U_i will denote the actual utilization of stream S_i , which can vary in the range $[U_i^{min}, U_i^{max}]$. To compute the average power P_i consumed by node i , we first 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} with bandwidth-guaranteed traffic.
- is in listening mode when the other nodes are transmitting¹, and it goes on sleep during idle times.

To take switching overhead into account, we assume 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, we assume that both t_i^{sw} and P_i^{sw} are constant and independent from the modes).

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

Let P_i^{tx} , P_i^{rx} , and P_i^{sl} be the power consumed in transmission mode, receiving mode, and sleep mode, respectively. The average power consumed by node i can be written as a weighted sum of the powers consumed by the node on 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. Thus, we have:

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

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

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

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

and gathering the terms,

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

Now, we define:

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

Hence, Equation (2) can be written as

$$P_i = U_i P_i^{ts} + \sum_{j=1, j \neq i}^n U_j P_i^{rs} + \Delta_i \quad (3)$$

From Equation (3), 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} ; that is, the lower the contribution given by the node utilizations on the power consumption. The worst case happens when $P_i^{tx} \simeq P_i^{rx} \simeq P_i^{sl}$, 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 sufficiently large t the energy $E_i(t)$ consumed in the worst case by node i at time t is:

$$E_i(t) = \int_0^t P_i(t) dt \simeq P_i \cdot t. \quad (4)$$

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 [15], [16].

In this work we consider the network lifetime as the time of the first node failure 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:

$$L_{net} = \min_i(L_i) = \min_i \left(\frac{E_i^0}{P_i} \right). \quad (5)$$

Analyzing the energy model expressed by Equation (3), it should be clear that the model 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 I). 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 energy into account saving 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 an info is available, the node can better esteem its energy consumption reducing the energy wasted for overhearing. For instance, if \hat{S} is the set of streams that do not send messages to node i , such a node can improve its energy model as follows:

$$P_i = U_i P_i^{ts} + \sum_{j=1, j \neq i, S_j \notin \hat{S}}^n U_j P_i^{rs} + \Delta_i. \quad (6)$$

In this example we considered that stream $S_j \in \hat{S}$ either always sends messages to node i , or it 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:

$$\begin{aligned}
 P_i^{rs}(t_{id} - 2t_{sw}) &\geq 2P_i^{sw}t_{sw} \\
 P_i^{rs} &\geq P_i^{sw} \frac{2t_{sw}}{t_{id} - 2t_{sw}} \quad (7)
 \end{aligned}$$

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.

C. Problem formulation

To design our algorithm, the following assumptions are envisaged. At any time, each node is able to monitor its available energy. To save its energy, a node i reduces its transmission power P_i^{tx} at 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 sense multi access (CSMA) is that nodes do not waste energy to contend the channel access and there are not packets 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 \mathbf{A} represents the largest utilization $U^*(\mathbf{A})$ of the network such that, for any real-time stream set whose total network utilization is $U \leq U^*(\mathbf{A})$, \mathbf{A} can guarantee the timeliness of each single real-time message.

To simplify the notation, when it is not necessary to specify the protocol, we will write U^* 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, we extend the definition of feasibility by also considering the system lifetime.

Definition III.1 Given a stream set $\Gamma = (S_1, S_2, \dots, S_n)$ 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 (3) we can observe that, to reduce the energy consumption, besides reducing the transmission powers P_i^{tx} , we could also increase the sleep time, by reducing the utilization of each node. In accordance with the elastic model presented in Section III-A, we assume that 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 $\Gamma = (S_1, S_2, \dots, S_n)$;
- The desired network lifetime L_{net}^d ;
- The worst case achievable utilization (WCAU) U^* of the MAC protocol;
- The vector of the initial available energies $\vec{E}^0 = (E_1^0, E_2^0, \dots, E_n^0)$;
- The transmission power vector $\vec{P}^{tx} = (P_1^{tx}, P_2^{tx}, \dots, P_n^{tx})$;
- The receiving power vector $\vec{P}^{rx} = (P_1^{rx}, P_2^{rx}, \dots, P_n^{rx})$;
- The sleep power vector $\vec{P}^{sl} = (P_1^{sl}, P_2^{sl}, \dots, P_n^{sl})$;
- The switching power vector $\vec{P}^{sw} = (P_1^{sw}, P_2^{sw}, \dots, P_n^{sw})$.

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:

$$\begin{aligned} \max : \quad & U = \sum_{i=1}^n U_i \\ \text{s.t. :} \quad & P_i \leq \frac{E_i^0}{L_{net}^d} \quad \forall i = 1, \dots, n \\ & U_i^{min} \leq U_i \leq U_i^{max} \quad \forall i = 1, \dots, n \\ & U \leq U^* \end{aligned} \quad (8)$$

IV. 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 paper.

Definition IV.1 Given a stream set $\Gamma = (S_1, S_2, \dots, S_n)$ with utilization U and lifetime L_{net} , the communication system is considered overloaded if $U > U^*$ or $L_{net} < L_{net}^d$.

A. The Elastic MAC algorithm

The El-MAC 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 $L_{net} \geq L_{net}^d$.

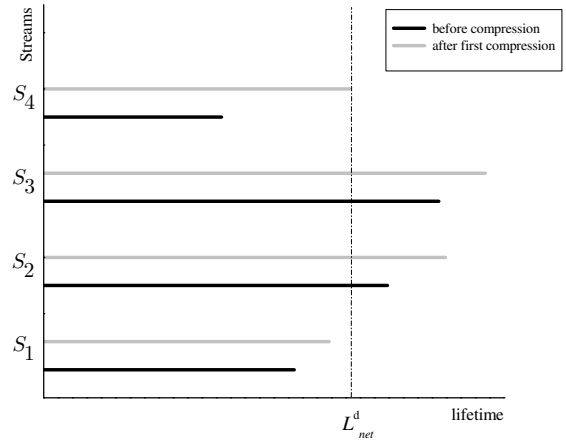


Figure 1. Example on the nodes' lifetime.

If the overload is caused by $U > U^*$, the elastic algorithm proposed in [14], can be used to compress the utilizations U_i so that $U = U^*$. If lifetime is also met under this setting, the algorithm stops returning the utilization vector $\vec{U} = (U_1, U_2, \dots, U_n)$. Otherwise, when $L_{net} < L_{net}^d$ if S_y is the stream with the shortest lifetime, all stream utilizations are reduced to increase L_y up to L_{net}^d . A sample scenario is illustrated in Figure 1, where, before compression, streams S_1 , and S_4 cannot reach the desired lifetime and S_1 is the stream with the shortest lifetime less than L_{net}^d , which determines the network lifetime.

To guarantee the lifetime constraint of the network we have to reduce the power consumption of the overloaded streams (S_1, S_4). Considering the dependencies expressed in Equation (3), 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 $L_4 = L_{net}^d$.

Notice that, after such a compression, all stream lifetimes will increase, but the lifetime of some streams could still be less than L_{net}^d . For instance, in the example shown in Figure 1, stream lifetimes after the first compression are indicated by the gray bars, and lifetime of stream S_1 is still less than L_{net}^d . In this condition, the stream with the shortest lifetime is S_1 , thus a new compression must take place to bring $L_1 = L_{net}^d$. In general, the compression procedure has to be repeated until the network lifetime constraint is satisfied, that is, until $\forall i P_i \leq E_i^0 / L_{net}^d$.

Notice that, at each compression step, stream utilizations are reduced using the elastic algorithm presented in [14], according to which the new utilizations are computed as follows:

$$U_i = U_i^0 - \frac{U^0 - U^d}{\varepsilon} \epsilon_i \quad (9)$$

where U^d is the total desired utilization, U_i^0 is the nominal utilization of stream S_i , U^0 is the sum of nominal utilizations, and $\varepsilon = \sum_{i=1}^n \epsilon_i$. The way to choose the elastic coefficients

is discussed in Section V.

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

In the following, the stream power expressed by Equation (3) is combined with Equation (9) to derive the value of U^d to be used as a reference in the compression algorithm.

From the energy model (Equation (3)):

$$\begin{aligned} \Delta P_y &= P_y^0 - P_y^d \\ &= (U_y^0 - U_y) P_y^{ts} + \sum_{j \neq y} (U_j^0 - U_j) P_y^{rs} \quad (10) \\ &+ \Delta_y^0 - \Delta_y. \quad (11) \end{aligned}$$

Using Equation (9), we can substitute $(U_y^0 - U_y)$ and $(U_j^0 - U_j)$ in Equation (10) as follows:

$$\begin{aligned} \Delta P_y &= \left(\frac{U^0 - U^d}{\varepsilon} \varepsilon_y \right) P_y^{ts} + \sum_{j \neq y} \left(\frac{U^0 - U^d}{\varepsilon} \varepsilon_j \right) P_y^{rs} \\ &+ \Delta_y^0 - \Delta_y \\ \Delta P_y &= \frac{U^0 - U^d}{\varepsilon} (\varepsilon_y P_y^{ts} + \sum_{j \neq y} \varepsilon_j P_y^{rs}) + \Delta_y^{old} - \Delta_y \\ \Delta P_y &= \frac{U^0 - U^d}{\varepsilon} (\varepsilon_y P_y^{ts} + (\varepsilon - \varepsilon_y) P_y^{rs}) + \Delta_y^0 - \Delta_y \\ \Delta P_y &= \frac{U^0 - U^d}{\varepsilon} (\varepsilon_y P_y^{tr} + \varepsilon P_y^{rs}) + \Delta_y^0 - \Delta_y \quad (12) \end{aligned}$$

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

From Equation (12) it follows that:

$$\frac{U^0 - U^d}{\varepsilon} = \frac{\Delta P_y - (\Delta_y^0 - \Delta_y)}{(\varepsilon_y P_y^{tr} + \varepsilon P_y^{rs})}. \quad (13)$$

Substituting Equation (13) into Equation (9), we have:

$$U_i = U_i^0 - \frac{\Delta P_y - (\Delta_y^0 - \Delta_y)}{\varepsilon_y P_y^{tr} + \varepsilon P_y^{rs}} \varepsilon_i. \quad (14)$$

Notice that, being $\Delta_y^0 - \Delta_y$ a small positive quantity, it can safely be neglected still leading to a conservative result. In this case, Equation (14) becomes:

$$U_i = U_i^0 - \frac{\Delta P_y}{\varepsilon_y P_y^{tr} + \varepsilon P_y^{rs}} \varepsilon_i. \quad (15)$$

where ΔP_y is given by:

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

The overall stream compression algorithm is shown in Figure 2. Summarizing, EI-MAC works as follows. The input parameter is the desired lifetime L_{net}^d . The two loops from

```

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

Figure 2. Stream compression algorithm.

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 EI-MAC. First, the index of the stream with lowest lifetime is obtained (line 10), then the desired utilization U^d is computed according to Equation (13). U^d is passed to the elastic compression procedure defined in [14]. After that, the new system lifetime is calculated and the cycle is repeated until either the lifetime constraint is met ($L_{net} \geq L_{net}^d$) or the system results to be infeasible ($\sum U_i^{min} > U^d$).

The complexity of the algorithm in Figure 2 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.

B. Applicative example

To clarify the proposed approach, we describe how to compute the parameters of the energy model (Equation 3), considering the RI-EDF protocol [7], as an example. RI-EDF is MAC layer protocol for cellular structured networks which 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 packet level, which is considered an atomic unit. 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 co-channel interferences are avoided. Moreover, in each cell any node can listen to any other. Without any loss of generality, we assume each cell being composed by $n - 1$ nodes plus a router node which is in charge of inter-cell communication. To each node, including the router, is associated with an elastic stream of periodic messages. For further details on the protocol see the original papers [5], [7].

We can remark that:

- 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 hyper-period H (i.e., the least common multiple of the stream periods). Thus, we have: $U_i^{sw} = (N_i^{sw}/H)t_i^{sw}$.
- 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, e.g. 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 3. 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:

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

We note that, when node 1 goes on transmission, node 2 goes on receiving mode and viceversa. Furthermore, both nodes go on sleep when the dummy node, associated with stream S_{Sl} , “access the channel”. From these last observations, we note that $N_1^{sw} = N_2^{sw} = N^{sw}$. Moreover, to calculate N^{sw}

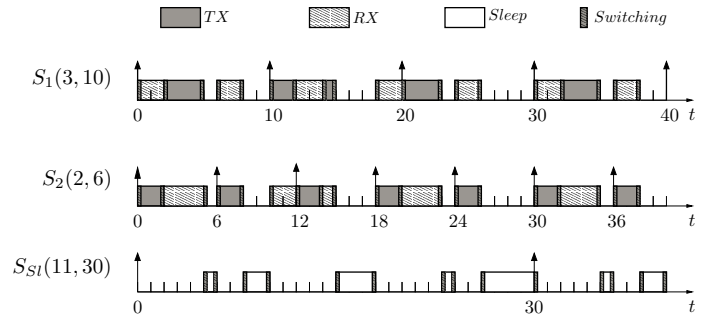


Figure 3. U_i^{sw} calculation with RI-EDF.

it is sufficient to count how many times each node, included the sleep node, accesses the channel during the hyperperiod $H = 30$. From Figure 3, we see node 1 accesses the channel 4 times, both node 2 and the sleep node 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 IV-A we assume that $\Delta_y^0 - \Delta_y = 0$, hence with EI-MAC it is not strictly necessary to calculate U_i^{sw} and U_i^{ctrl} .

V. ALGORITHM PROPERTIES

In this section we report some observations and results on the EI-MAC properties. First, it is worth observing that in general, the proposed algorithm (EI-MAC) provides a solution for a more relaxed problem than that defined at the end of Section III (refer to optimization problem (8)). In other words, EI-MAC provides an utilizations vector $\vec{U} = (U_1, U_2, \dots, U_n)$, 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;
- $U \leq U^{opt}$;

where U^{opt} is the solution of the optimization problem (8). The algorithm provides a solution such that the stream set Γ is feasible in the sense of Definition III.1, but this solution might not be the optimal one.

Observe that, if the system lifetime is met when for every i $U_i = U_i^{min}$, EI-MAC 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.

A. Properties of the algorithm parameters

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

For the sake of clarity, in the remaining part of the paper we consider to have homogenous nodes (nodes with the same radio transceiver), that is, $P_i^{rs} = P^{rs}$, $P_i^{ts} = P^{ts}$ for all i . We also consider $\Delta_i \simeq \Delta$ for all i .

The next lemma is used to prove the subsequent theorem.

Lemma V.1 Given a message stream set $\Gamma = \{S_1, S_2, \dots, S_n\}$, 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: From the energy model, $P_i \geq P_v$ means that:

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

then

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

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

$$(U_i - U_v) P^{ts} \geq (U_i - U_v) P^{rs} \quad (17)$$

From Inequality (17), 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 $U_i \leq U_v$.

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, i.e. the cycle (from line 9 to line 19 in Figure 2) is executed just once.

Theorem V.2 Given an elastic stream set $\Gamma = \{S_1, S_2, \dots, S_n\}$, 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-MAC provides a solution by just one iteration if and only if for all i :

$$\epsilon_y \leq \epsilon_i \frac{1 + k_i}{1 - k_i} + \frac{k_i}{1 - k_i} \epsilon^{i,y} \quad (18)$$

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

$$\epsilon_y \geq \epsilon_i \frac{1 + k_i}{1 - k_i} + \frac{k_i}{1 - k_i} \epsilon^{i,y} \quad (19)$$

$$\text{where } \epsilon^{i,y} = \left(\sum_{j=1, j \neq i, y}^n \epsilon_j \right) = \left(\sum_{j=1}^n \epsilon_j \right) - \epsilon_i - \epsilon_y.$$

Proof: First we prove the Inequality (18).

(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 V.1, $\forall i$ 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 (9),

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

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

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

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

$$\begin{aligned} (\epsilon_y - \epsilon_i) &\leq k_i \left(\left(\sum_{j \neq i, y} \epsilon_j \right) + \epsilon_i + \epsilon_y \right) \\ \epsilon_y (1 - k_i) &\leq \epsilon_i (1 + k_i) + k_i \epsilon^{i,y} \\ \epsilon_y &\leq \epsilon_i \frac{1 + k_i}{1 - k_i} + \frac{k_i}{1 - k_i} \epsilon^{i,y} \end{aligned}$$

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

Now, suppose that the algorithm converges by one iteration, hence $P_y^d = E_i^0 / L_{net}^d$ and $\forall i P_y^d \geq P_i$. To prove the Inequality (19) it is sufficient to observe that, from Lemma V.1, if $P^{ts} \leq P^{rs}$ then $P_y^d \geq P_i$ if and only if $U_y \leq U_i$, that is, $U_y - U_i \leq 0$. The rest of the prove is similar to that for the Inequality (18), where it is sufficient to exchange the signs (\leq) and (\geq).

It is worth noticing that, from Inequality (18), if for some i $k_i > 1$, $P^{ts} \geq P^{rs}$ then $\epsilon_y < 0$. This means that, we have no solutions because ϵ_y must be non negative. Further, from Inequality (19), if $k_i = 1$ and $P^{ts} \leq P^{rs}$ we have 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 above are impossible.

Corollary V.3 Given an elastic stream set $\Gamma = \{S_1, S_2, \dots, S_n\}$, 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: We start defining a function $f_\epsilon(k_i) = \epsilon_i \frac{1+k_i}{1-k_i} + \frac{k_i}{1-k_i} \epsilon^{i,y}$, where $i \neq y$, and $\epsilon_i, \epsilon^{i,y}$ are given parameters.

In the first case, where $P^{ts} \geq P^{rs}$, from Lemma V.1, at each step the algorithm chooses y so that for all i , $U_y^0 \geq U_i^0$, then $k_i \geq 0$. we can observe that:

- From Theorem V.2, if $k_i < 1$ then to converge by just one iteration we need that $\epsilon_y \leq f_\epsilon(k_i) < \infty$. For $k_i = 1$, $f_\epsilon(k_i) = \infty$. Thus, we can choose $\epsilon_y \leq f_\epsilon(k_i) \leq \infty$ for $0 \leq k_i \leq 1$.
- If $k_i > 1$, from Inequality (18), to converge by just one iteration we need that $\epsilon_y \leq f_\epsilon(k_i) < 0$. Note that, being ϵ_y an elastic coefficient, $\epsilon_y < 0$ has no meaning. However, it is not difficult to see that if for all i , $0 \leq \epsilon_y \leq \epsilon_i$ then $U_y \geq U_i$, that is: the algorithm finishes by just one iteration. Therefore, being the Inequality (18) a necessary and sufficient condition, it follows that either $0 \leq k_i \leq 1$ or Inequality (18) is incorrect. Since we proved the correctness this latter inequality, the first part of the thesis follows.

In the second case we have $P^{ts} \leq P^{rs}$. From Lemma V.1 at each step the algorithm chooses y so that for all i , $U_y^0 \leq U_i^0$, this means that $k_i \leq 0$.

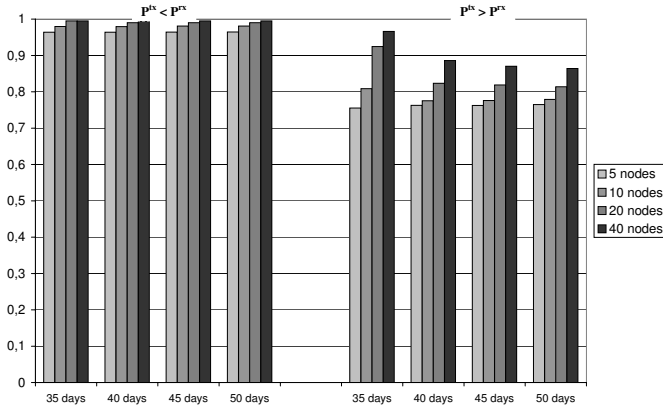


Figure 4. El-MAC performance.

From Theorem V.2, in order to get the coefficients ϵ_i , it is necessary to solve a linear inequalities system. However, as the function $f_\epsilon(k_i) = \epsilon_i \frac{1+k_i}{1-k_i} + \frac{k_i}{1-k_i} \epsilon^{i,y}$ is increasing monotonic for $0 \leq k_i < 1$ and decreasing monotonic for $k_i < 0$, once we chose each ϵ_i except ϵ_y , defined $k_{min} = \min_i(k_i)$ and $k_{max} = \max_i(k_i)$, it is sufficient to choose ϵ_y as indicated by the Inequalities (18, 19), solving just one inequality, considering only k_{min} and k_{max} respectively. In this way, the relations of the Theorem V.2 can be satisfied without solving a linear inequalities system.

VI. SIMULATION RESULTS

To test the algorithm we performed several simulation experiments varying the main input parameters. We tested El-MAC varying the desired system lifetime (measured on days), the number of nodes involved in the network, and the transceivers used. In particular, we have considered two communication devices, the *CC2400* [1] and *DR3300* [2], mainly for the fact that the former has $P^{tx} < P^{rx}$ and the latter has $P^{tx} > P^{rx}$. To calculate P^{tx} and P^{rx} we have obtained the operating voltage and the currents sunk in the various working modes from their data sheets.

The following upper bound is used to evaluate the algorithm. From the energy model, expressing U_i as function of P_i and U_j , for all $i = 1, \dots, n$ and $j = 1, \dots, n$ with $j \neq i$, we can derive a necessary condition on U :

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

Summing up for all i both side of the equation, and remembering that $U = \sum_i U_i$, we have:

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

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

$$U = \frac{P_{tot} - n\Delta}{P^{ts} + (n-1)P^{rs}}. \quad (20)$$

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, we define the following channel utilization upper bound:

$$U^{ub} = \frac{P_{tot}^{max} - n\Delta}{P^{ts} + (n-1)P^{rs}}. \quad (21)$$

To meet L_{net}^d , a necessary but not sufficient condition is that $P_{tot} \leq P_{tot}^{max}$. From Equations (20) (21), it can be seen that condition $P_{tot} \leq P_{tot}^{max}$ is equivalent to condition $U \leq U^{ub}$. In this way, we found out an upper bound for U^d . In other words, let U^{opt} be the solution of Problem (8), it turns out that $U^d \leq U^{opt} \leq U^{ub}$.

The algorithm has been evaluated calculating the performance ratio p_{ratio} between the total utilization U^d , provided by El-MAC, and the theoretical upper limit U^{ub} :

$$p_{ratio} = \frac{U^d}{U^{ub}}. \quad (22)$$

The simulation was performed generating, for each stream S_i , the utilizations U_i^{max} and U_i^{min} using the method proposed by Bini et al. [17]. The maximum total utilization, i.e. the sum of the U_i^{max} , was equal to U^* for each simulation run. In our simulations, we considered a MAC protocol having $U^* = 1$; an example of such a protocol is RI-EDF (see Section II).

Usually, in the elastic model the coefficients ϵ_i represent the relative importance of each stream. For instance, some traffic streams can be compressed a little because are important for the system operation, some other could be less important hence they can be compressed more. In general, the importance of a stream depends on the application hence, since the evaluation of El-MAC is done without considering any particular application, the elastic coefficients ϵ_i have been generated as random numbers with uniform distribution in $[0,1]$.

As mentioned above, each experiment set was built selecting three parameters: the number of nodes n , the desired system lifetime L_{net}^d , and the transceiver. For each experiment set, the algorithm was simulated 1000 times, generating U_i^{max} and U_i^{min} for each run, providing the average value of p_{ratio} . The algorithm has been implemented using MATLAB [18].

Figure 4 shows on the left side the results obtained with the transceiver *CC2400*, having $P^{tx} < P^{rx}$. The same figure shows on the right side the results with the transceiver *DR3300*, having $P^{tx} > P^{rx}$. The figure shows the performance ratio under different desired system lifetimes (days) and number of nodes n . Experiments are grouped by L_{net}^d measured in days, and each element in a group represents the performance ratio for a specific number of nodes in the network.

As we can see, for $P^{tx} < P^{rx}$, fixing n and increasing L_{net}^d , p_{ratio} presents a very small variation. Instead, by fixing L_{net}^d and varying n , p_{ratio} grows up to 1. In other words, from the definition of the performance ratio, by increasing the number of nodes the algorithm produces a solution closer to the channel upper bound U^{ub} ; that is, the solution tends to be optimal. Increasing the desired lifetime, in practice the algorithm performance does not change. It is worth noticing that, when $P^{tx} < P^{rx}$, the performance ratio is very close to 1 in all simulated scenarios.

For the case $P^{tx} > P^{rx}$, for n equal to 20 and 40, varying L_{net}^d the performance ratio decreases. This last remark is not valid for n equal to 5 and 10, because in this case p_{ratio} remains more or less the same when L_{net}^d changes.

Fixing L_{net}^d , the performance ratio p_{ratio} increases with n . In the second scenario ($P^{tx} > P^{rx}$), the solutions provided by the algorithm are characterized by a lower performance ratio with respect to the first setting in which $P^{tx} < P^{rx}$.

VII. CONCLUSION AND FUTURE WORK

In this paper we analyzed the problem of saving energy at the communication level for a distributed embedded system, formed by a network of wireless nodes. First, we described the main sources of energy waste regarding a wireless network composed by distributed devices. We 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 given interval. Given the available energy on each node, we derived an energy model that allows estimating the network lifetime. 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, we introduced an algorithm (EI-MAC) to select the bandwidth of each stream within a specified range, so that both real-time constraints and the network lifetime can be guaranteed. We also showed that, if the elastic coefficients of the streams are appropriately chosen, the algorithm ends by just one iteration.

We tested EI-MAC by simulation, using the parameters provided by the datasheets of two different radio transceivers. The first one with $P^{tx} > P^{rx}$ and the second one with $P^{tx} < P^{rx}$. We used as performance metric a theoretic upper bound on the channel utilization U^{ub} , which is an upper bound for a solution of the optimization problem (8), introduced in Section III-C.

The proposed energy model and EI-MAC are directly applicable to TDMA scheduling approaches, as for instance the IEDF protocol. The major differences among the various scheduling protocols can reside in the computation of U^{sw} and U^{ctrl} , which strictly depend on the MAC protocol adopted.

As a future work, we plan to devise an algorithm able to produce a solution to the optimization problem described in Section III-C through the Equation (8).

REFERENCES

- [1] *Chipcon CC2420*, <http://www.ti.com>.
- [2] *Rfm DR3300*, <http://www.rfm.com>.
- [3] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transaction on networking*, vol. 12, no. 3, pp. 493–506, June 2004.
- [4] M. Adamou, I. Lee, and I. Shin, "An energy efficient real-time medium access control protocol for wireless ad-hoc networks," in *Proc. of the 22th IEEE Real-Time Systems Symposium (RTSS 2001)*.
- [5] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo, "An implicit prioritized access protocol for wireless sensor networks," in *Proc. of the IEEE Real-Time Systems Symposium (RTSS 2002)*, Dec. 2002.
- [6] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, Feb. 1973.
- [7] T. L. Crenshaw, A. Tirumala, S. Hoke, and M. Caccamo, "A robust implicit access protocol for real-time wireless collaboration," in *Proc. of the IEEE Euromicro Conference on Real-Time Systems (ECRTS 2005)*, Jul. 2005.
- [8] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher, "SPEED: A stateless protocol for real-time communication in sensor networks," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS 2003)*.
- [9] J. A. Stankovic, T. He, T. F. Abdelzaher, M. Marley, G. Taoand, S. Son, and C. Lu, "Feedback control scheduling in distributed systems," in *Proc. of the 22th IEEE Real-Time Systems Symposium (RTSS01)*, London, UK, Dec. 2001.
- [10] A. Koubaa, M. Alves, and E. Tovar, "Energy/Delay Trade-off of the GTS Allocation Mechanism in IEEE 802.15.4 for Wireless Sensor Network," *International Journal of Communication Systems*, vol. 20, no. 7, pp. 791–808, July 2007.
- [11] A. Rowe, R. Mangharam, and R. Rajkumar, "RT-Link: A Time-Synchronized Link Protocol for Energy-Constrained Multi-hop Wireless Networks," in *Third IEEE Int. Conf. on Sensors, Mesh and Ad Hoc Communications and Networks (IEEE SECON06)*.
- [12] D. D. Demarch and L. B. Becker, "An integrated scheduling and retransmission proposal for firm real-time traffic in IEEE 802.11e," in *Proc. of the IEEE Euromicro Conference on Real-Time Systems (ECRTS 2007)*, Jul. 2007.
- [13] M. M. Sobral and L. B. Becker, "A wireless hybrid contention/tdma based mac for real-time mobile application," in *Proc. of the ACM Symposium on Applied Computing (SAC 2008)*, March. 2008.
- [14] G. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, "Elastic scheduling for flexible workload management," *IEEE Transactions on Computers*, vol. 51, no. 3, pp. 289–302, March 2002.
- [15] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Communication Letters*, vol. 9, no. 11, pp. 976–978, November 2005.
- [16] D. M. Blough and P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks," in *Proceedings of MOBICOM 2002*.
- [17] E. Bini and G. Buttazzo, "Biasing effects in schedulability measures," in *Proc. of the 16th Euromicro Conference on Real-Time Systems (ECRTS 2004)*, June 2004.
- [18] *MATLAB*, <http://www.mathworks.com>.