

RESEARCH

Open Access



# DECA: distributed energy conservation algorithm for process reconstruction with bounded relative error in wireless sensor networks

Felipe da Rocha Henriques<sup>1\*</sup>, Lisandro Lovisolo<sup>2</sup> and Marcelo Gonçalves Rubinstein<sup>2</sup>

## Abstract

When a wireless sensor network (WSN) is employed to monitor environmental data, efficient energy usage is demanded so that the WSN lifetime can be extended. In this context, we propose a distributed energy conservation algorithm (DECA). DECA considers the known past to make predictions on the measurement change over time in order to put sensor nodes into sleep mode, saving energy. Sensor nodes compute their inactivity periods taking into account the inactivity periods of their neighboring nodes. The signals are reconstructed at the sink node, from the sampled values received from sensor nodes. DECA is designed to keep the reconstruction error of the monitored process below a given fraction of its actual value. Since DECA works on the application layer, it can be combined with medium access control and routing protocols to further improve energy management. The proposed algorithm is evaluated with simulations using real data. Results show gains in the network lifetime of up to 1000 % as compared to a WSN without any energy management, while maintaining the error of the reconstructed process at the sink within the desired fraction imposed. Since a strategy for managing the sleeping period in routing nodes is proposed, a metric for evaluating the effects of sleeping periods on network connectivity is also proposed. We have verified that the network connectivity is not impaired by the proposed algorithm, as packet delivery ratios above 90 % were obtained.

**Keywords:** Connectivity, Energy, Estimation, Reconstruction, Wireless sensor networks

## 1 Introduction

Recent advances in micro-electronics and wireless communications made it possible to develop and deploy low cost, low energy consumption, and tiny sensors. These have been used to build wireless sensor networks (WSNs) [1]. A WSN can be applied in several domains [2–4], such as (i) in medical applications, to remotely monitor patients and their biometric data; (ii) for military purposes, to monitor forces; (iii) in industrial automation; and (iv) to sense variables in a region of interest.

In this work, we consider that a WSN is employed to sense a physical variable, a field—a process whose value depends on space coordinates  $(x, y, z)$  and on time  $t$ . Each

sensor  $S_i$  samples the monitored process at its position  $(x_i, y_i, z_i)$  measuring  $s(x_i, y_i, z_i, t)$  and, eventually, transmits the measurement to a sink node (for the sake of simplicity in the reminder, we ignore the coordinate  $z$  and use the notation  $s_i(t) = s(x_i, y_i, t)$ ). In a WSN, the sink is usually more robust than the sensor nodes, and it can be used as a gateway [1].

The problem that we consider is: How can we make an energy-efficient usage of the WSN while providing an acceptable reconstruction of the sensed field for the monitoring application? As acceptable reconstruction, we assume that the reconstruction error is kept below a pre-defined value, while energy efficiency involves improving the network autonomy, by increasing its lifetime. More specifically, we assume that the network lifetime is the time until the first node dies, i.e., when its energy ends [5].

In this work, we propose a distributed energy conservation algorithm (DECA) that runs in the application layer

\*Correspondence: henriquesfelipe@telpet.com.br

<sup>1</sup>Coordenação de Telecomunicações, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ), Petrópolis, Brasil  
Full list of author information is available at the end of the article

for wireless sensor networks in monitoring applications. Roughly, each sensor independently considers information which is expected to be available at the sink node to predict a future measured value and this in turn is used to compute the inactivity period (the interval while the node stays in sleep mode) of a given node  $S_i$ . Each sensor node sleeps during its inactivity period, and then the node wakes up, measures the desired quantity, and transmits it if the error between the current measurement and the predicted behavior of the monitored field is larger than a given threshold, reiterating the process. Energy saving and the resulting increase in network lifetime is achieved by both reducing the amount of transmissions and also by putting nodes to sleep during their inactivity periods.

The decision whether or not to sleep is performed by each node individually, that is locally. Nodes do not depend on the reception of sleeping commands or data from the sink. Therefore, when making this decision, two aspects need to be considered. The first is that for computing its inactivity period (IP<sub>*i*</sub>), each node  $S_i$  must consider how the sink reconstructs the process from the measurements the sink receives. As a consequence, in order to guarantee a trusty reconstruction of the process, we impose the constraint of keeping the reconstruction error within an acceptable distortion criterion at the sink and not solely at the sensor. Therefore, since each node takes its own decision, the proposed strategy is decentralized and distributed.

An important issue to be considered when computing the inactivity/sleeping period of a sensor is: How does putting a given node to sleep impact the network connectivity? Therefore, the second aspect driving DECA design refers to multi-hop networks, where nodes can act both as sources (when they measure samples of the monitored process) and as routers/relays (when besides measuring they also forward packets from neighbors), that is, the inactivity period of a sensor node must be at least considered by a sensor that is routing its messages. Inversely, when deciding its inactivity period, each node must account for the impacts of this decision on network connectivity, that is, on other nodes. This is considered in DECA; nodes forward their own inactivity periods together with the measurements and an inactivity period is used by the first-hop router when it computes its inactivity period.

Therefore, although running in the application layer and the decision of how long to sleep being taken autonomously by each sensor, network topology and routes awareness are inherent to DECA since the inactivity periods of sensors being routed by a node are considered by the node when deciding for how long to sleep. In addition, a fringe benefit of forwarding inactivity periods is that sensor nodes are not required to be synchronized,

since the algorithm does not demand an absolute time base, employing just relative times (time intervals).

We should highlight that DECA differs from existing proposals in the literature as (i) DECA is applied in the application layer; (ii) DECA does not impose an uniform sampling interval for the measurements time-series; (iii) by design, DECA imposes the reconstruction error of the sensed variable to be within an acceptable distortion criterion; (iv) DECA is decentralized and distributed, since each and every node decides by itself for how long to sleep, although considering information about the nodes it forwards packets for; and (v) DECA uses a deterministic approach for computing the node sleeping/inactivity period.

Since nodes are put to sleep during their estimated inactivity periods, in order to evaluate the impact of this in the network connectivity, we propose a metric that we call success ratio.

The proposed algorithm is assessed by simulations. The IEEE 802.15.4 network standard, which is widely used in WSNs, is considered in the simulations. In addition real environmental data are employed. Results show a significant increase in the network lifetime by using DECA, as compared to WSNs without any energy management scheme. We have also verified that network connectivity is not impaired by the algorithm, what, indeed, makes it worth to use DECA on any monitoring application alike the one considered by DECA.

This work is structured as follows. Section 2 presents related works and proposals dealing with energy conservation in WSNs. In Section 3, we present a model for WSN based monitoring applications. In Section 3.2.1, this model is used to size-up the impacts of putting nodes to sleep in WSNs, and it is employed to derive the proposed algorithm for saving energy in monitoring applications (DECA). In Section 4, we describe DECA algorithm implementation. In Section 5, we present the sensor node energy model used in this work and the simulation aspects. In Section 6, we present results obtained with simulations considering real environmental data. At last, conclusions are discussed in Section 7.

## 2 Energy saving in WSNs

The design of node's hardware, its miniaturization, and energy harvesting have gained importance for WSN deployment, but, at the same pace, methods leading to energy saving in a WSN have also turned into an important issue. In [6], a survey of energy saving methods for WSNs is presented, including a taxonomy for energy saving schemes. Some energy saving methods are briefly mentioned hereafter.

According to [1], communication (i.e., transmission and reception) is the task that requires more energy in a WSN. This means that it may be advantageous for a node to

process data, in order to compress it or to decide whether to transmit it or not and thus save energy while sleeping [7]. In [8], spatial and temporal correlations between measured samples are used to decrease the amount of transmissions, saving sensor node energy. The sink predicts the field being sensed by sensor nodes using information that it receives from the nodes. Using the received data, the sink estimates for how long each node can sleep and then sends different messages for the distinct nodes conveying their sleeping periods. This strategy is centralized as the sink decides for how long the nodes can sleep, and a given node may not be put to sleep at all, in the case that packets are lost, thus severely impacting energy saving. Differently, our strategy is completely decentralized, since each sensor node computes its own sleeping period.

Similarly, in [9], the temporal pattern of samples measured by sensor nodes is used to reduce the amount of transmissions to the sink node. The monitored process is compared against its expected behavior, if they match then the node does not transmit its value in order to increase network lifetime. If the measurements do not match the expected behavior, then the nodes transmit the values. Since this scheme does not involve node sleeping, a confinement of its gain in network lifetime is obtained. DECA tries to track the monitored process and the sensor nodes do not need to know (a priori) the behavior of the process that is being sensed. Moreover, DECA makes node to switch into a sleeping mode, increasing the autonomy of sensor nodes.

In [10], a sleep-awake scheme is proposed, in which a network coordinator periodically transmits a beacon frame with a sleeping command. Therefore, this scheme is centralized and nodes enter the sleeping state synchronously upon the reception of this command. To improve energy saving, a power control mechanism in the MAC sublayer is also proposed, based on the distance between neighboring nodes. In our proposal, sensor nodes do not receive any sleeping command (which could eventually be lost because of the wireless channel). Furthermore, DECA runs at application layer of sensor nodes, independently of lower-layer protocols.

In [3], a “derivative-based prediction” is employed in each sensor node to manage the need for transmissions. It simply allows each node to verify if the measured quantity did not change enough, in which case it is not transmitted. This approach resembles the one presented in [11]. The energy conservation algorithm in [11] aims at reducing the amount of transmissions by managing the need for them. A sample is transmitted only if the percentage variation between it and the last transmitted sample is greater than a given threshold and nodes sleep between transmissions. The current proposal outperforms the previous one [11] because DECA obtains a constrained reconstruction error.

The proposals in [3, 8, 9] consider a uniform sampling interval for the sampled measurements. This is similar to the proposals for “signal compression” based on piecewise linear smoothing [12]. However, once a node is put into sleep mode, the sampling is not uniform anymore; this aspect emerged in [11] and is fully considered by the algorithm presented in this work.

Finally, DECA can be used together with other energy saving protocols, as for example the ones proposed to save energy in the network layer and in the MAC sublayer [13–18] in order to further augment energy saving.

### 3 Sensing and reconstructing a process with a WSN

Figure 1 summarizes the problem of using a WSN to sense a field. Sensor nodes are spread in a region of interest and measure a collection of samples  $s_i^{(M)}$  of a physical quantity— $i$  indicates the sensor and  $(M)$  is used to indicate that these values are in principle known only by the measurement node. As sensor nodes use an algorithm to save energy by managing the necessity of communication, a subset of their measurements is transmitted. The set of transmitted samples by node  $i$  is  $s_i^{(T)}$ . From node  $i$ , the sink node receives the set  $s_i^{(R)}$ . Since a wireless channel is used, some packets may be lost and therefore  $s_i^{(R)} \subset s_i^{(T)}$ . From the received samples, the sink reconstructs the field.

#### 3.1 Measurements sampling, transmission, and reception

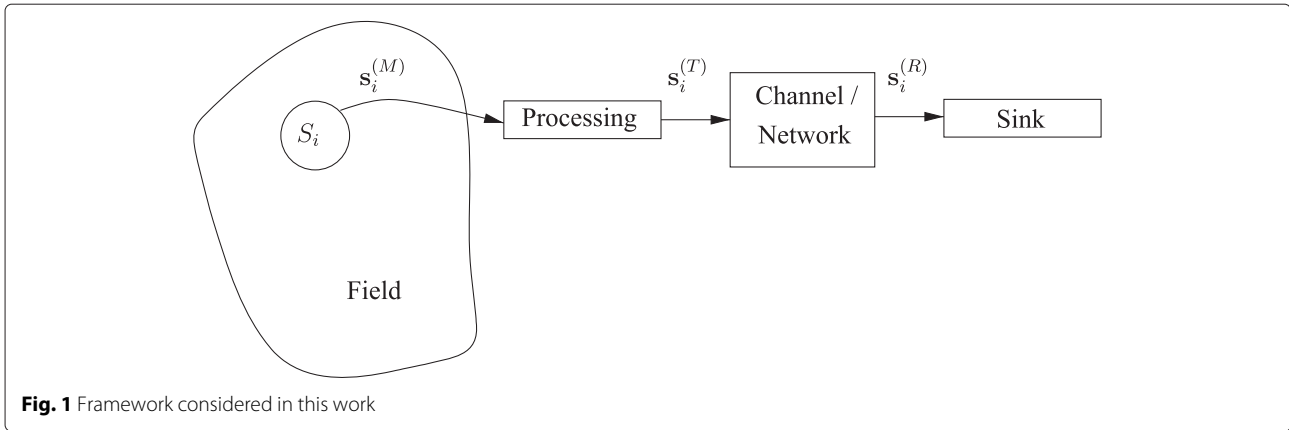
Let us assume that:

- A measured sample  $s_i^{(M)}$  may be transmitted or not, i.e., it may generate a transmitted sample  $s_i^{(T)}$  or not, therefore  $s_i^{(T)} \subset s_i^{(M)}$
- A transmitted sample  $s_i^{(T)}$  may be received or not, i.e., it may generate a received sample  $s_i^{(R)}$  or not, therefore  $s_i^{(R)} \subset s_i^{(T)}$
- The monitored process is considered to be sufficiently smooth to be tracked with a simple predictor
- The received samples are equal to the transmitted samples, except by eventually lost data

Consider a sensor node  $S_i$  and a sink node  $S_s$ . The measurements collected by  $S_i$  can be organized as a vector

$$\mathbf{s}_i^{(M)} = \left[ s_i^{(M)}(t_i^{(M)}[1]), s_i^{(M)}(t_i^{(M)}[2]), \dots, s_i^{(M)}(t_i^{(M)}[N]) \right], \quad (1)$$

where the  $t_i^{(M)}[n]$ ,  $n \in [1 \dots N]$  are the measurement/sampling times and are such that  $t_i^{(M)}[j] <$



$t_i^{(M)}[k]$  if  $j < k$ . One can also define the vector composed of the instants when the elements of  $\mathbf{s}_i^{(M)}$  are measured

$$\begin{aligned} \mathbf{s}_i^{(M)} &= [s_i^{(M)}[1], s_i^{(M)}[2], \dots, s_i^{(M)}[N]], \text{ and} \\ \mathbf{t}_i^{(M)} &= [t_i^{(M)}[1], t_i^{(M)}[2], \dots, t_i^{(M)}[N]]. \end{aligned} \quad (2)$$

At the sink node, the reconstruction of the process is done from the available information, which is in principle a subset of the measured samples. Since  $\mathbf{s}_i^{(R)} \subset \mathbf{s}_i^{(T)} \subset \mathbf{s}_i^{(M)}$ , reconstruction employs the received samples  $\mathbf{s}_i^{(R)}$  and the corresponding times  $\mathbf{t}_i^{(R)}$ , i.e.,

$$\begin{aligned} \mathbf{s}_i^{(R)} &= [s_i^{(R)}[1], s_i^{(R)}[2], \dots, s_i^{(R)}[K]] \text{ and} \\ \mathbf{t}_i^{(R)} &= [t_i^{(R)}[1], t_i^{(R)}[2], \dots, t_i^{(R)}[K]]. \end{aligned} \quad (3)$$

The values of the elements of  $\mathbf{t}_i^{(R)}$  are the measurement times of samples that are transmitted by the sensors and that are received by the sink. At any given time of the network life the cardinality of  $\mathbf{s}_i^{(R)}$  is smaller than or equal to the cardinality of  $\mathbf{s}_i^{(M)}$ , that is  $K \leq N$ . One notes that  $\mathbf{t}_i^{(R)} \subset \mathbf{t}_i^{(M)}$  and that  $\mathbf{s}_i^{(R)} \subset \mathbf{s}_i^{(M)}$ . Although, the mapping from  $n$  in  $s_i^{(M)}[n]$  to  $k$  in  $s_i^{(R)}[k]$  will be considered in Section 3.3.

### 3.2 Reconstruction of the process

Let  $s_i(t) = s(x_i, y_i, t)$  be the value collected by sensor  $S_i$  placed at  $(x_i, y_i)$  over time and its reconstruction version at the sink be  $\hat{s}_i(t) = \hat{s}(x_i, y_i, t)$ . The reconstruction error can be defined as

$$e(S_i, t) = e(x_i, y_i, t) = s_i(t) - \hat{s}_i(t). \quad (4)$$

We want to reconstruct the process  $s(x, y, t)$  within an acceptable error criterion. The reconstruction  $\hat{s}_i(x, y, t)$  can be obtained using an interpolator.

#### 3.2.1 Interpolator data

The sink forms the set of received samples from node  $S_i$ ,  $\mathbf{s}_i^{(R)}$ . In order to save energy nodes do not transmit all measured samples, i.e.  $\mathbf{s}_i^{(T)} \subset \mathbf{s}_i^{(M)}$ . In addition, some transmissions may be lost, therefore,  $\mathbf{s}_i^{(R)} \subset \mathbf{s}_i^{(T)} \subset \mathbf{s}_i^{(M)}$ . For the moment, we assume that  $\mathbf{s}_i^{(R)} \equiv \mathbf{s}_i^{(T)}$ ; packet loss impact is discussed in Section 6.

At time  $t$ , the set measurements collected by  $S_i$  is

$$\mathbf{s}_i^{(M)}(t) = \left\{ s_i^{(M)}(t_i^{(M)}[j]) \right\}_{t_i^{(M)}[j] < t} = \left\{ s_i^{(M)}[j] \right\}_{t_i^{(M)}[j] < t}. \quad (5)$$

This set allows to predict  $s_i^{(M)}(t)$  using a causal predictor  $\hat{s}_i^{(M)}(t) = \mathcal{P} \left\{ \mathbf{s}_i^{(M)}(t) \right\}$ .

The prediction can also employ the instants of sample measurements,  $\mathbf{t}_i^{(M)}(t) = \left\{ t_i^{(M)}[j] \right\}_{t_i^{(M)}[j] < t}$ , turning into

$$\hat{s}_i^{(M)}(t) = \mathcal{P} \left\{ \mathbf{s}_i^{(M)}(t), \mathbf{t}_i^{(M)}(t) \right\}. \quad (6)$$

Replacement of the superscript  $(M)$  by  $(R)$  provides a predictor that can be applied at the sink node. It is important to note that although the sets  $\mathbf{s}_i^{(R)}(t)$  and  $\mathbf{t}_i^{(R)}(t)$  are defined similarly to  $\mathbf{s}_i^{(M)}(t)$  and  $\mathbf{t}_i^{(M)}(t)$ , they are not the same, since every measurement is neither transmitted by the sensor nor received by the sink.

#### 3.2.2 Error criterion

We impose the ratio between the predicted sample and its actual value to be smaller than a fraction  $\delta$  of the actual value. This criterion at the node and at the sink translates as

$$\begin{aligned} \left| \hat{s}_i^{(M)}(t) - s_i^{(M)}(t) \right| &\leq \delta \left| s_i^{(M)}(t) \right| \text{ and} \\ \left| \hat{s}_i^{(R)}(t) - s_i^{(R)}(t) \right| &\leq \delta \left| s_i^{(R)}(t) \right|. \end{aligned} \quad (7)$$

Assuming the use of interpolators, the errors of the estimated field in the sensor node and at the sink at a given instant  $t$  are given by

$$e\left(s_i^{(M)}, t\right) = \left| \mathcal{P} \left\{ \mathbf{s}_i^{(M)}(t), \mathbf{t}_i^{(M)}(t) \right\} - s_i^{(M)}(t) \right|, \text{ and} \quad (8)$$

$$e\left(s_i^{(R)}, t\right) = \left| \mathcal{P} \left\{ \mathbf{s}_i^{(R)}(t), \mathbf{t}_i^{(R)}(t) \right\} - s_i^{(R)}(t) \right|, \quad (9)$$

while the error bounds in Eq. (7) impose

$$e\left(s_i^{(M)}, t\right) \leq \delta \left| s_i^{(M)}(t) \right| \text{ and } e\left(s_i^{(R)}, t\right) \leq \delta \left| s_i^{(R)}(t) \right|. \quad (10)$$

Sensor nodes may estimate a future measurement  $\hat{s}_i^{(M)}[n] = \hat{s}_i^{(M)}\left(t_i^{(M)}[n]\right)$  from some known samples and their measurement times. Similarly, we can use the time interval between  $\hat{s}_i^{(M)}[n]$  and  $s_i^{(M)}[n-1]$ , given by  $\hat{t}_i^{(M)}[n] - t_i^{(M)}[n-1]$ , for the node to sleep and save energy. Putting this altogether, using the restriction  $e\left(s_i^{(M)}, t\right) \leq \delta \left| s_i^{(M)}(t) \right|$ , we can estimate the inactivity period/sleeping time of node  $i$ . This resembles the approach on “signal compression” in [12]; however, our approach neither imposes nor assumes a uniform sampling ( $t_i[n] = n\Delta T$ ).

The problem gets more interesting when one imposes that the reconstruction (R) at the sink must closely match the measured (M) process. This is described by

$$\left| \hat{s}_i^{(R)}(t) - \hat{s}_i^{(M)}(t) \right| \leq \delta \left| \hat{s}_i^{(M)}(t) \right|, \quad (11)$$

that is, we impose the reconstructed value at the sink to be bounded by the error criterion with respect to the measurement prediction at the sensor.

Considering that measurements are always positive (without any loss of generality), from Eq. (11), two situations arise: (i) If  $\hat{s}_i^{(R)}[n] > \hat{s}_i^{(M)}[n]$ , then  $\hat{s}_i^{(R)}[n] - \hat{s}_i^{(M)}[n] \leq \delta \hat{s}_i^{(M)}[n] \Rightarrow \hat{s}_i^{(R)}[n] \leq \hat{s}_i^{(M)}[n](1 + \delta)$  and (ii) else if  $\hat{s}_i^{(R)}[n] \leq \hat{s}_i^{(M)}[n]$ , then  $-\hat{s}_i^{(R)}[n] + \hat{s}_i^{(M)}[n] \leq \delta \hat{s}_i^{(M)}[n] \Rightarrow \hat{s}_i^{(R)}[n] \geq \hat{s}_i^{(M)}[n](1 - \delta)$ . These situations can be easily combined into

$$1 - \delta \leq \frac{\hat{s}_i^{(R)}[n]}{\hat{s}_i^{(M)}[n]} \leq 1 + \delta. \quad (12)$$

The parameter  $\delta$  defines the error bounds (and may be application dependent), as we show that it allows to estimate for how long sensors can be put in sleeping mode.

### 3.2.3 First order predictor

A first order model for  $\mathcal{P}\{\cdot\}$  employs the last two samples to estimate a future value: as the last two measured samples and instants of measurements, we have  $s_i^{(M)}[n-1]$  and  $s_i^{(M)}[n-2]$ ,  $t_i^{(M)}[n-1]$ , and  $t_i^{(M)}[n-2]$ . We use  $\hat{t}_i^{(M)}[n]$  to denote an instant that has not passed yet, that is esti-

ated. Using these definitions, the predicted value (see Fig. 2) is given by

$$\begin{aligned} \hat{s}_i^{(M)}[n] &= \mathcal{P} \left\{ \mathbf{s}_i^{(M)}\left(\hat{t}_i^{(M)}[n]\right), \mathbf{t}_i^{(M)}\left(\hat{t}_i^{(M)}[n]\right) \right\} = \quad (13) \\ &= s_i^{(M)}[n-1] + \frac{s_i^{(M)}[n-1] - s_i^{(M)}[n-2]}{t_i^{(M)}[n-1] - t_i^{(M)}[n-2]} \\ &\quad \times \left( \hat{t}_i^{(M)}[n] - t_i^{(M)}[n-1] \right). \end{aligned}$$

Assuming that all transmissions are successful, i.e.,  $\mathbf{s}_i^{(R)} \equiv \mathbf{s}_i^{(T)}[k]$ , a similar equation holds for  $\hat{s}_i^{(R)}[k]$  (the impact of breaching this assumption is evaluated in Section 6).

We can define the differences (a) between the two last measured samples, (b) between their sample times, (c) between the two last transmitted samples, and (d) between their sample times, using

$$\Delta s_i^{(M)}[n-1] = s_i^{(M)}[n-1] - s_i^{(M)}[n-2], \quad (14)$$

$$\Delta t_i^{(M)}[n-1] = t_i^{(M)}[n-1] - t_i^{(M)}[n-2], \quad (15)$$

$$\Delta s_i^{(R)}[k-1] = s_i^{(R)}[k-1] - s_i^{(R)}[k-2], \text{ and} \quad (16)$$

$$\Delta t_i^{(R)}[k-1] = t_i^{(R)}[k-1] - t_i^{(R)}[k-2]. \quad (17)$$

From these, we can define the variation rates

$$\begin{aligned} \alpha_i^{(M)}[n-1] &= \frac{\Delta s_i^{(M)}[n-1]}{\Delta t_i^{(M)}[n-1]} \text{ and} \\ \alpha_i^{(R)}[k-1] &= \frac{\Delta s_i^{(R)}[k-1]}{\Delta t_i^{(R)}[k-1]}. \end{aligned} \quad (18)$$

Using the above notations, the first order prediction rules become

$$\begin{aligned} \hat{s}_i^{(M)}[n] &= s_i^{(M)}[n-1] + \alpha_i^{(M)}[n-1] \\ &\quad \left( \hat{t}_i^{(M)}[n] - t_i^{(M)}[n-1] \right) \text{ and} \end{aligned} \quad (19)$$

$$\begin{aligned} \hat{s}_i^{(R)}[k] &= s_i^{(R)}[k-1] + \alpha_i^{(R)}[k-1] \\ &\quad \left( \hat{t}_i^{(R)}[k] - t_i^{(R)}[k-1] \right). \end{aligned} \quad (20)$$

A future sample is guessed using linear interpolation which requires very low computational complexity and memory requirements. As depicted in Fig. 2, this rule could be used to estimate for how long the sensor node can sleep at most. Nevertheless, one wants to keep the reconstruction error at the sink within the bounds in Eq. (11). Therefore, the influence of the quantities known at the sink on the sensor decision to sleep or not needs to be addressed.

### 3.3 How does information cognized at the sink influence sensor nodes?

All information that is available at the sink about the measurements of a sensor is known by the sensor itself. On

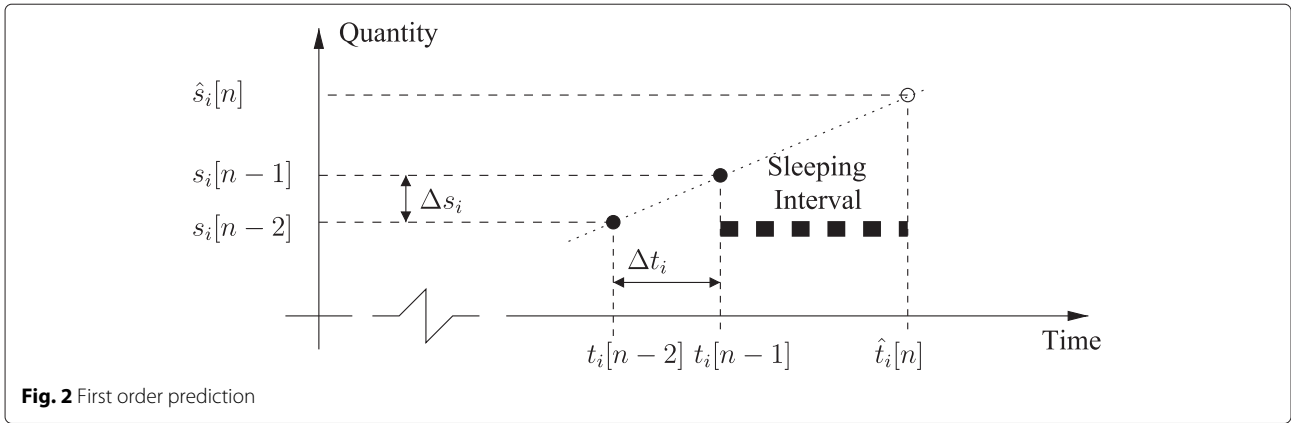


Fig. 2 First order prediction

the other hand, the contrary is hardly true. Let us resort the indices of the last two transmitted (received) samples and their instants of transmissions with respect to the sample indices at the sensor. In doing so, we have  $s_i^{(R)}[n-j]$  (a measurement taken  $j$  samples before the current one) and  $s_i^{(R)}[n-(j+l)]$  ( $j+l$  samples before the current) and their corresponding measurement times  $t_i^{(R)}[n-j]$  and  $t_i^{(R)}[n-(j+l)]$ . The lags  $j$  and  $j+l$  indicate that not all measured samples are transmitted to (nor received at) the sink. Using this indexing strategy one has that  $\Delta s_i^{(M)}[n-1] = s_i[n-1] - s_i[n-2]$ ,  $\Delta t_i^{(M)}[n-1] = t_i[n-1] - t_i[n-2]$ ,  $\Delta s_i^{(R)}[n-1] = s_i[n-j] - s_i[n-j-l]$ , and  $\Delta t_i^{(R)}[n-1] = t_i[n-j] - t_i[n-j-l]$ .

The variation rates in Eq. (18) are now given by

$$\alpha_i^{(M)}[n-1] = \frac{s_i[n-1] - s_i[n-2]}{t_i[n-1] - t_i[n-2]} \text{ and } \alpha_i^{(R)}[n-1] = \frac{s_i[n-j] - s_i[n-j-l]}{t_i[n-j] - t_i[n-j-l]} \quad (21)$$

Therefore, we have that Eqs. (19) and (20) can be expressed as

$$\hat{s}_i^{(R)}[n] = s_i[n-j] + \alpha_i^{(R)}[n-1] (\hat{t}_i[n] - t_i[n-j]) \text{ and} \quad (22)$$

$$\hat{s}_i^{(M)}[n] = s_i[n-1] + \alpha_i^{(M)}[n-1] (\hat{t}_i[n] - t_i[n-1]). \quad (23)$$

The predictions in Eqs. (22) and (23) are illustrated in Fig. 3. Equation (22) estimates the next measurement (of node  $S_i$ )  $\hat{s}_i^{(R)}[n]$  to be received at the sink, from the last received measurements  $s_i[n-j]$  and  $s_i[n-j-l]$  and their times. Meanwhile, Eq. (23) provides an estimate of the next sample to be collected  $\hat{s}_i^{(M)}[n]$  from the two last,  $s_i[n-1]$  and  $s_i[n-2]$  and their times. Equation (23) is applicable only at the sensor and it describes the “local” behavior of the first order predictor while the data employed to compute  $\hat{s}_i^{(R)}[n]$  is available at the sensor and also at the sink.

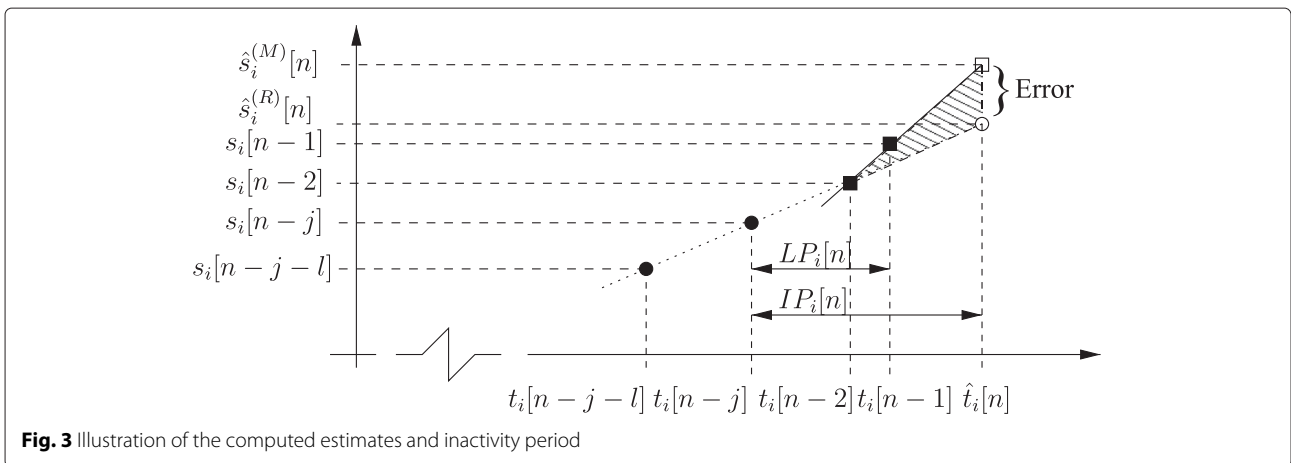


Fig. 3 Illustration of the computed estimates and inactivity period

### 3.3.1 Inactivity period computation

Let us define the time interval between the last transmitted sample and the next sample to be collected ( $IP_i[n]$ ) and the time interval between the last transmitted sample and the last measured sample ( $LP_i[n]$ ) of node  $S_i$ , i.e.,

$$IP_i[n] = \hat{t}_i[n] - t_i[n - j], \text{ and} \quad (24)$$

$$LP_i[n] = t_i[n - 1] - t_i[n - j]. \quad (25)$$

With these definitions, Eqs. (22) and (23) become

$$\hat{s}_i^{(R)}[n] = s_i[n - j] + \alpha_i^{(R)}[n - 1] IP_i[n], \text{ and} \quad (26)$$

$$\hat{s}_i^{(M)}[n] = s_i[n - 1] + \alpha_i^{(M)}[n - 1] (IP_i[n] - LP_i[n]). \quad (27)$$

Replacing Eqs. (26) and (27) in Eq. (12) to guarantee the desired error bound, after simple manipulations, we obtain the two inequalities

$$IP_i[n] \leq \frac{s_i[n - j] - s_i[n - 1] (1 - \delta) + LP_i[n] [\alpha_i^{(M)}[n - 1] (1 - \delta)]}{\alpha_i^{(M)}[n - 1] (1 - \delta) - \alpha_i^{(R)}[n - 1]}, \text{ and} \quad (28)$$

$$IP_i[n] \geq \frac{s_i[n - j] - s_i[n - 1] (1 + \delta) + LP_i[n] [\alpha_i^{(M)}[n - 1] (1 + \delta)]}{\alpha_i^{(M)}[n - 1] (1 + \delta) - \alpha_i^{(R)}[n - 1]}. \quad (29)$$

The sensor is aware of its transmissions to the sink. Therefore, one applies the sink reconstruction model at the sensor for obtaining an acceptable range for its inactivity period,  $IP_i[n]$ . This range is guaranteed by choosing the smallest case between Eqs. (28) and (29). The acceptable range of  $IP_i[n]$  for a given sensor node  $i$  guarantees a reconstruction error within a prescribed error at the sink. This range depends on (i) the relationship of the variation rates of the sensor at the sink  $\alpha_i^{(R)}[n - 1]$  and at the sensor itself  $\alpha_i^{(M)}[n - 1]$ ; (ii) the distortion bound  $\delta$ ; and (iii) the time interval between the last transmitted sample and the last measured sample,  $LP_i[n]$ .

Considering the use of a simple linear predictor, we derived for how long a node can sleep while the reconstruction is within an allowed distortion range. For that each sensor compares the predictor model available at the sink for its measurements against a predictor model for the sensor, which is more frequently updated, i.e., it considers non-transmitted measurements. It is worth noticing that other more sophisticated interpolators as spline, cubic, or polynomial could also be evaluated. Next, we use this linear predictor to propose DECA.

## 4 Distributed energy conservation algorithm for wireless sensor networks – DECA

DECA works directly in the application layer of each sensor node, regardless, for example, of the routing protocol used. The steps of the algorithm are showed in

Algorithm 1. The algorithm is presented for a node  $S_i$  and considers the sensor energy  $\mathcal{E}_i$  and that the sensor runs until its energy ends. That is, at each step of the algorithm,  $\mathcal{E}_i$  is decreased accordingly to the task that is performed (see Section 5.1). The inactivity period is initially set to  $T_{\text{granularity}}$  seconds and it is updated by DECA at each sensor. After that, the sensing–processing–(eventual) transmission–sleeping procedure occurs. For ease of understanding, Algorithm 1 is presented in two parts: the first one describes the initialization of sensor nodes (this takes two sampling to have the measurement and transmission queues filled (lines 4–19); the second one is the “main loop” (lines 20–35), in which nodes transmit

---

### Algorithm 1: DECA running at node $S_i$ .

---

```

// INITIALIZATION:
1  n ← 1;
2  p ← 1;
3  IP_i[n] ← T_granularity;
4  while n ≤ 2 do
5      measure s_i^{(M)}[n] at t_i^{(M)}[n]
      (s_i^{(M)}[n] = S_i^{(M)}(t_i^{(M)}[n]));
6      if n = 2 then
7          IP_i[n] ← minimum between Eqs. (28) and (29);
8          if there are packets to forward from neighbors then
9              forward the data from the neighbors;
10             IP_i[n] ← min(IP_i[n], {IP_k[n]}_{k ∈ SN_i});
11         end
12     end
13     s_i^{(T)}[p] ← s_i^{(M)}[n];
14     t_i^{(T)}[p] ← t_i^{(M)}[n];
15     transmit s_i^{(T)}[p] and t_i^{(T)}[p] together with IP_i[n];
16     sleep for γIP_i[n] seconds;
17     p ← p + 1;
18     n ← n + 1;
19 end
// MAIN LOOP:
20 while E_i > 0 do
21     measure s_i^{(M)}[n] at t_i^{(M)}[n]
      (s_i^{(M)}[n] = S_i^{(M)}(t_i^{(M)}[n]));
22     IP_i[n] ← minimum between Eqs. (28) and (29);
23     if |s_i^{(M)}[n] - s_i^{(M)}[n]| / s_i^{(M)}[n] > δ then
24         if there are packets to forward from neighbors
25             then
26                 forward the data from the neighbors;
27                 IP_i[n] ← min(IP_i[n], {IP_k[n]}_{k ∈ SN_i});
28             end
29             s_i^{(T)}[p] ← s_i^{(M)}[n];
30             t_i^{(T)}[p] ← t_i^{(M)}[n];
31             transmit s_i^{(T)}[p] and t_i^{(T)}[p] together with IP_i[n];
32             p ← p + 1;
33         end
34         sleep for γIP_i[n] seconds;
35         n ← n + 1;
36 end

```

---

only if the error between the current measurement and the predicted behavior of the monitored field is larger than allowed.

In Algorithm 1,  $n$  indexes the current measurement ( $s_i^{(M)}[n]$  at  $t_i^{(M)}[n]$ ) and  $p$  the current transmission ( $s_i^{(T)}[p]$  at  $t_i^{(T)}[p]$ ). We use these two distinct indices because some measurements are not be transmitted. Once a node wakes up, it measures the quantity and compares the sampled value against the predicted one, then the node transmits the new value only if the error is larger than  $\delta$ . This is a basic procedure for saving energy, since unnecessary transmissions are avoided. Following, the sensor node updates its inactivity period and sleeps again.

The sleeping period reduction factor  $\gamma$  ( $0 < \gamma \leq 1$ ) is used to increase the probability of  $S_i$  to be awake to forward packets from its neighbors. Experiments evaluating the impact of  $\gamma$  on DECA are presented in Section 6.2. More details on how DECA considers neighbors (when the sensor has to route data from other sensors) and the reasons for the use of  $\gamma$  are presented in the following.

**4.1 Neighbors, routing, and their inclusion in DECA**

A given node may act as router for others. The sleeping period of node  $S_i$  must consider the impact it may provoke on the transmission of the neighbor sensors that use it as a router. In lines 8–11 and 24–27 of Algorithm 1, the node verifies if there are packets to be forwarded. If not, then the inactivity period  $IP_i[n]$  depends solely on the quantities that the sensor measures and the ones already available at the sink for reconstruction. Otherwise, the actual inactivity period of the node must consider also the sleeping period of its neighbors.

Therefore, for router nodes, the inactivity period  $IP_i$  of each network node  $S_i$  must consider the inactivity periods

of its neighbors (nodes that use  $S_i$  as a router), because  $S_i$  has to be awake to forward packets from them. DECA considers that by making the inactivity period of a sensor  $S_i$  to be defined by

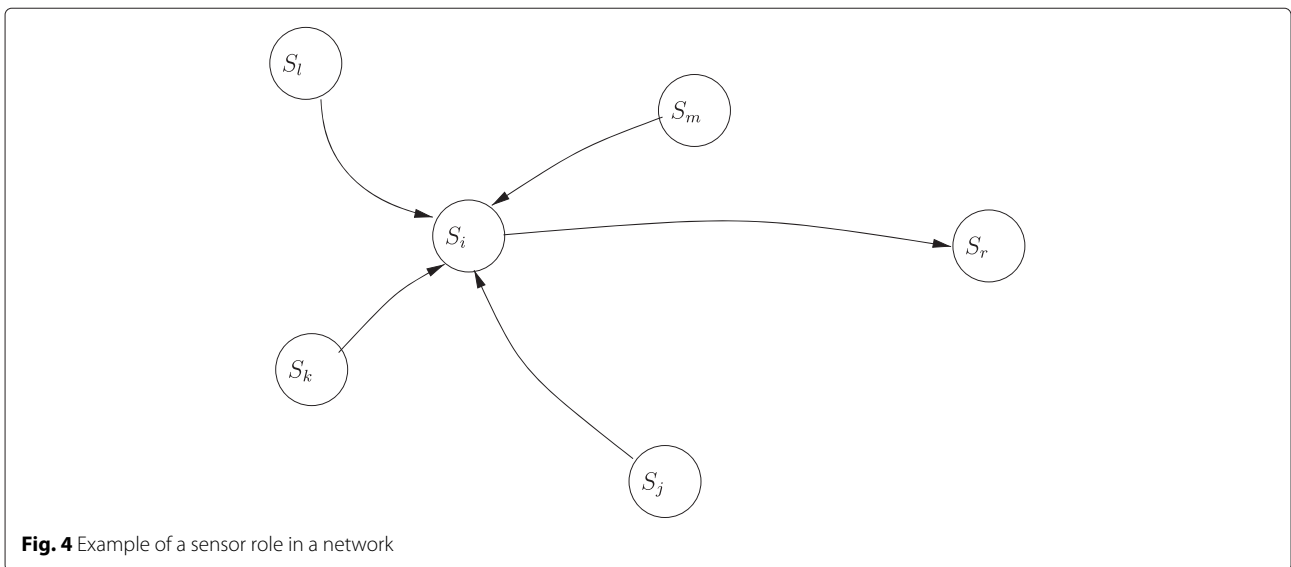
$$IP_i[n] = \min (IP_i[n], \{IP_k[n]\}_{k \in SN_i}), \tag{30}$$

where each  $IP_k[n]$  represents the inactivity period of each neighbor of  $S_i$  and  $SN_i$  represents the set of neighbors of  $S_i$ . In this work, Eq. (30) is applied at each sensor node in the WSN, since any of them can act as a router. The impact and effectiveness of such an strategy are evaluated in following sections. Figure 4 exemplifies a node scenario with its neighboring nodes; node  $S_i$  is assumed to have  $\#SN_i$  neighboring nodes (in the example, there are four neighboring nodes— $S_j, S_k, S_l$ , and  $S_m$ —whose packets are routed by node  $S_i$ ).

**4.2 Some comments on DECA and its parameters**

DECA aims at tracking the variation rate of the monitored process. Nodes should sleep for shorter periods when the monitored process varies more rapidly because they have to transmit more packets. Otherwise, nodes tend to sleep for larger periods when the variation of the monitored process reduces.

DECA has a very low complexity, as each node  $S_i$  basically computes the bounds on  $IP_i$  from Eqs. (28) and (29) and chooses the smallest value. If the node also acts as router then it computes the minimum among that bound and the inactivity periods received from neighbors. Each sensor needs to store just the last two measurements of the process and their corresponding measurements instants as well as the two last transmitted measurements



**Fig. 4** Example of a sensor role in a network



and corresponding measurements instants. In the case the node is a router, it is also required to store a list with the inactivity periods of nodes that use it as a router.

As discussed, DECA employs three parameters: (1) an initial sampling interval ( $T_{granularity}$ ) or inactivity period (which is adjusted locally in each sensor regarding what it measures); (2)  $\delta$ , the acceptable maximum fraction of distortion; and (3) the parameter  $\gamma$  used to augment the probability of the node to be awoken to forward packets from its neighbors.

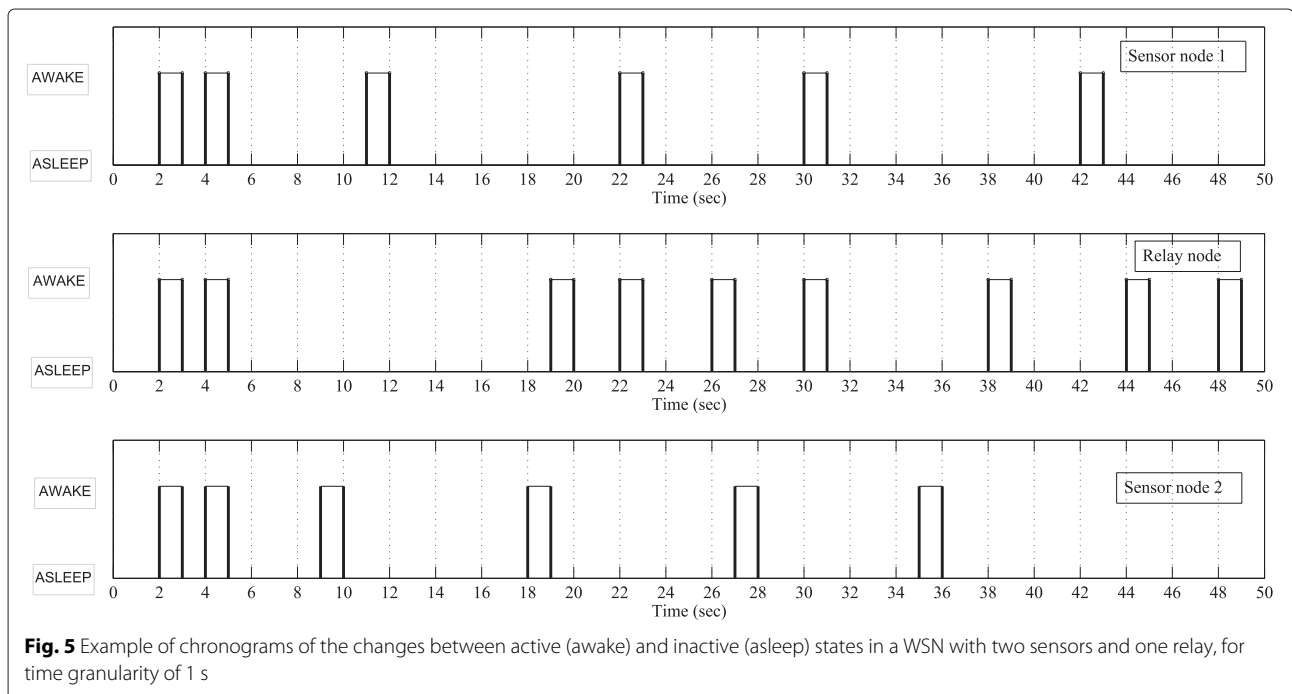
The parameter  $T_{granularity}$  is the initial sampling interval. It is adjusted by DECA as time goes by. The inactivity period is initially defined to be equal to  $T_{granularity}$ . As the nodes sleep during their inactivity periods, samples are taken when the nodes wake up, and therefore it is inherent to DECA sampling the process at a non-uniform sampling rate whose elapsed time between samples is adaptively adjusted. One should note that this time lapse is not the period of the sensor clock, which, in actual deployments, for an awake sensor, is orders of magnitude shorter than this time lapse between successive samples. The parameter  $\delta$  is defined considering the maximum percentage error allowed in the reconstruction process. The parameter  $\gamma$  is also defined for DECA deployment. The influences of both on DECA are thoroughly evaluated in the sequel.

### 4.3 Some comments on sensors clocks and sampling times

Synchronization of sensor nodes would require at least some message exchange among them, consuming energy.

Therefore, sensor clocks may have small deviations from one to another. However, in our approach, sleeping intervals are computed neither in function of clock pulses nor using time instants. This somehow mitigates drift among sensors clocks. The sleeping interval is obtained from counting clock cycles; therefore, mismatches among sleeping periods are smaller than they would be if instead one considers sensor clock counts since their turning-on instants. Consequently, there is no large impairment on the network connectivity when altering sensor duty cycles as employed by DECA.

Figure 5 presents chronograms of sensor states (between awake or asleep) for nodes in a WSN. More specifically, the states in function of time for one relay and its two neighbors are presented as time goes by. The chronograms were obtained from a simulation run with time granularity of 1 s (sensors stay active during the granularity period). As presented in Algorithm 1, sensor nodes transmit their measurements when in the active state (if the variation of the measured quantity requires that). A relay listens, receives, and forwards transmissions from its neighbors and also acquires and transmits its own measurements. One notes that a sensor node may only transmit packets to its relay when in the active state. For the packet to be received, the relay must also be active. Consequently, only if there is an intersection (over time) between the active periods of the sensor node and its correspondent relay, the packet may be received (if there is no collisions or other reasons for outage in the physical layer). In addition, one readily notes from Fig. 5 that the



relay node sleeps for shorter periods than sensor ones, an expected from Eq. (30) and Algorithm 1.

## 5 Performance evaluation

### 5.1 Energy model

We assume that each node operates switching between two states: (i) inactive—an energy saving state (sleep mode) [7], in which the sensor radio interface is switched off or (ii) active. In active state, a node is in one of four operation modes: (a) measuring, (b) processing, (c) transmission, and (d) receiving. Therefore, a state-based energy model is adopted in this work. We have extended the energy model in [19]—an empirical model for TELOS commercial hardware—to consider the different operation modes.

The energy consumption of a node is estimated as a function of the period of time in which the node stays in the inactive and the active state and in the different operation modes. Therefore, one can use

$$\widehat{EC} = t_I \times C_I + t_A \times C_A + t_M \times (C_A + C_M) + t_P \times (C_A + C_P) + t_R \times (C_A + C_R) + t_T \times (C_A + C_T), \quad (31)$$

to evaluate energy consumption, in which  $t_I$ ,  $t_A$ ,  $t_M$ ,  $t_P$ ,  $t_R$ , and  $t_T$  are, respectively, the cumulative sum of intervals in which the node remains in inactive and active states, and in measuring, processing, receiving, and transmitting operation modes.

In the active state, a fixed consumption of 10 mJ/s is taken into account; this consumption does not consider performing any specific task. In the transmission mode, a linear relationship between energy consumption and the payload size of the transmitted packet was observed, while for reception, energy consumption was observed to be independent of payload size [20]. Moreover, a consumption of 0.034 mJ for transmitting 1 byte during 0.58 ms was observed [19]. Meanwhile, for reception and sleep modes, 62.4 and 1.8 mJ/s of consumption are reported [19], respectively. Summarizing, the associated consumptions to each state and mode of the energy model employed are presented in Table 1.

### 5.2 Simulation scenario

WSNs monitoring environmental data (temperature and humidity signals) are considered. Data comes from the WSN of the Intel Berkeley Research lab, where for more than a month, 54 nodes sensed environmental data [21, 22]. These data correspond to temperature and humidity signals (among others), which indeed depend on the sensor node coordinates  $x_i$  and  $y_i$  and on time  $t$  ( $i$  is the index of a given node  $S_i$ ); we use  $sT_i(t) = s_T(x_i, y_i, t)$  and  $sH_i(t) = s_H(x_i, y_i, t)$ , to refer to them, respectively.

**Table 1** Energy consumption parameters employed in the simulations

Node initial energy (J)	2.00
Transmission power (dBm)	−5
Reception sensibility (dBm)	−66
Radio range (m)	40
$C_I$ : Inactive state consumption (mJ/s)	1.80
$C_A$ : Active state consumption (mJ/s)	10.00
$C_M$ : Measuring mode consumption (mJ/s)	18.00
$C_P$ : Processing mode consumption (mJ/s)	18.00
$C_R$ : Rx mode cConsumption (mJ/s)	62.40
$C_T$ : Tx mode cConsumption (mJ/s)	58.62
Payload size (byte)	1

A multi-hop communication model is considered, using the ad-hoc on demand distance vector (AODV) routing protocol [23, 24], in order to forward packets hop-by-hop from sources to the sink. Initially, a WSN with 15 of the 54 nodes of the Berkeley WSN is considered. The sink node is located in the position of node  $S_{20}$  of [21], (0.5,17) in meters. Coordinates are relative to the upper right corner of the lab. In each simulation run, the positions of the 14 sensor nodes are drawn from the remaining 53 nodes.

The presented simulations consider as benchmark a WSN at which nodes sample and transmit the sensed variable at  $R_s = 1/T_{\text{granularity}}$ . DECA starts with the same value for sensors inactivity periods. However, this value is adjusted for each and every sensor, as samples are taken and packets are transmitted. In addition, sensors are considered to have an initial energy of 2J stored in their batteries.

We consider that all sensor values are correct, without noise. As the sensed process is reconstructed at sink, DECA imposes that the reconstruction error has to be kept within an acceptable distortion criterion, defined by  $\delta$ .

Simulations were performed using TrueTime 1.5 [25], an environment based in MatLab/Simulink, and the network standard was the IEEE 802.15.4 [26, 27]. The CSMA/CA MAC sublayer protocol with no beacons is employed, and therefore, packet losses can occur due to collisions. We also consider a log-distance path loss model as a propagation model.

### 5.3 Evaluation criteria

The criteria used to evaluate the results obtained with DECA are:

- Reconstruction error: This is defined in Eq. (4). In addition, to obtain a more precise evaluation of the reconstruction error, we employ the cumulative distribution function (CDF) of it in our analyses.

- **Transmission decrease:** This evaluates the reduction in the amount of transmissions (in %) as compared to a network without any energy management—for identical node positions and operation starting times.
- **Lifetime:** As pointed out in Section 1, we adopt as network lifetime the elapsed time until the first node exhaust its energy.
- **Lifetime increase:** This evaluates the increase of the network lifetime in percentage, by comparing the network lifetime using DECA against the lifetime of the network (for identical node positions and operation starting times) without any energy management scheme.
- **Packet delivery ratio:** This is the ratio between the amount of received packets and the amount of transmitted packets.
- **Success ratio:** This is the ratio between the amount of packets that a node  $S_i$  has forwarded to the sink  $S_r$  and the amount of packets that its neighbors forwarded to it, that is, the success ratio considers all packets that  $S_i$  has to forward, even if it is sleeping. Therefore, one defines

$$SR_i = \frac{\sum_{j=1}^N \text{forwarded packets}_{i,j \dots N}^j}{\sum_{j=1}^N \text{forwarded packets}_{j \dots N}^j}, \quad (32)$$

in which  $i$  is the index of a relay node  $S_i$ , and  $j \dots N$  are the indexes of each neighbor of  $S_i$ . This metric aims at evaluating the amount of packets that may have been lost by a given relay  $S_i$ .

The success ratio of  $S_i$  in forwarding packets from its neighbors may vary with its inactivity period  $IP_i$ . If  $S_i$  sleeps for a long period, in order to save energy, then a reduction in the success ratio may be observed due to possible packet losses. Then, a direct influence of  $\gamma$  on the success ratio is expected. Another relevant issue that must be investigated is the influence of the quantity of neighboring nodes  $\#SN_i$  that use  $S_i$  as relay node on the success ratio. One can expect the success ratio to be affected by  $\#SN_i$ . Figure 4 presents an example in which node  $S_i$  forwards packets from its neighbors  $SN_i = \{S_j, S_k, S_l, S_m\}$  to a sink node  $S_r$ . We employ such scenarios both to evaluate the success ratio and to analyze the network connectivity. More specifically, the effects caused by (1) the factor  $\gamma$  and (2) the amount of nodes using  $S_i$  as a router in both success ratio and packet delivery ratio are addressed.

## 6 Results

Initially, we consider a fifteen-node WSN used to sense temperature and humidity real data that uses a multi-hop communication to forward packets from sensor nodes to the sink. For the presented results,  $\gamma = 0.5$  (the effects of  $\gamma$  are investigated in Section 6.2) and threshold  $\delta$  values of 1, 2, 5, and 8 % are used. Each simulation scenario is run

ten times (randomly selecting the 15 nodes composing the WSN from the 54 possible ones), and 95 % confidence intervals for the mean are used when applicable in result graphs, where these intervals are represented by vertical bars.

Results are presented in four subsections, aiming at evaluating different aspects. In the first subsection, the objective is to evaluate the proposed algorithm in augmenting network lifetime, while keeping the error within a predefined fraction. In the second subsection, one aims at evaluating how relay nodes behave when executing the algorithm. In the third, it is investigated how the algorithm behaves in terms of network scalability; networks of different sizes are considered. In the fourth, one aims at investigating the impact of the time granularity in the simulations with DECA, so that possible bias in the evaluation criteria can be overcome.

### 6.1 Overall algorithm evaluation

Figure 6 presents the percentage reduction in the amount of transmissions and the network lifetime increase, in function of  $\delta$ , for the monitored environmental data: temperature and humidity. The percentages are computed with respect to networks without any energy management strategy; nodes take measures and transmit them periodically at  $T_{\text{granularity}}$ . We consider a  $T_{\text{granularity}}$  of 0.1 s. It should be noted that the actual sampling rate used in the WSN in [21] was 31 s. Here, we assume that the data from [21] is originally sampled at 0.1 s instead of 31 s, that is, as if the process were faster than it originally is. The results for other sampling rates are presented in Section 6.4. Figure 6 shows that increasing  $\delta$  leads to a reduction of transmissions and therefore augments the network lifetime. Increasing  $\delta$  makes nodes transmit measured samples with greater variation rates because a larger reconstruction error is allowed. Therefore, nodes may have larger sleeping periods and then transmit fewer measurements. Figure 6 shows gains in the network lifetime of up to 1000 % as compared to a WSN without any energy management.

Figure 7 shows the cumulative distribution function (CDF) of the reconstruction error of the monitored signals (temperature and humidity) for the different  $\delta$ . These curves allow to evaluate if the reconstruction error is kept within the desired threshold  $\delta$ . The increase of  $\delta$  leads to the increase of the largest reconstruction error. This behavior is in consonance with results presented in Fig. 6 and derives from the fact that as  $\delta$  increases, fewer samples are used to reconstruct the monitored signals. However, more important is to observe in Fig. 7 that the reconstruction error is always smaller than the set threshold ( $\delta$ ). In order to show that explicitly, Table 2 shows the maximum percentage reconstruction error, in function of the parameter  $\delta$ , for the two monitored fields. It can be verified

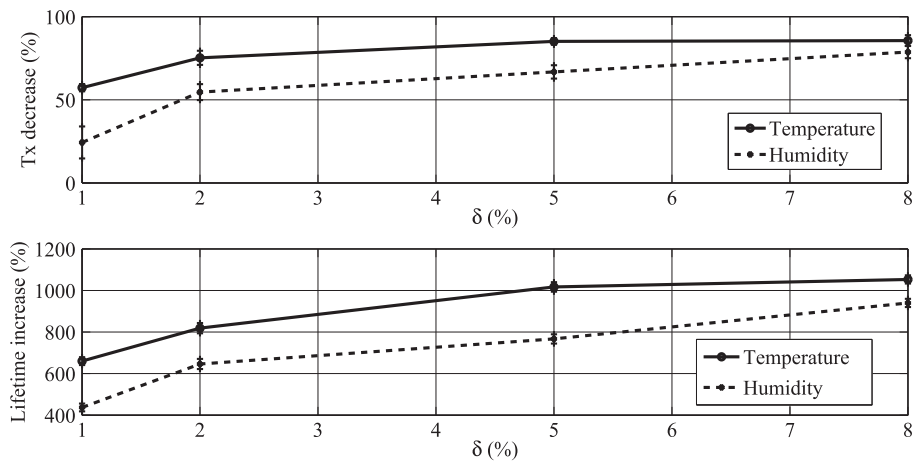


Fig. 6 Transmission decrease and lifetime increase  $\times \delta$

that the imposed constraint (reconstruction error within a desired fractional margin) is satisfied in all cases.

We can observe from Fig. 7 and from the data in Table 2 that the reconstruction error for humidity is greater than for temperature. In order to understand this, we evaluate the variation rate of the monitored signals. Using their partial derivatives, we have

$$sT'_i(t) = \frac{\partial sT_i(t)}{\partial t} \text{ and } sH'_i(t) = \frac{\partial sH_i(t)}{\partial t}. \quad (33)$$

This provides an idea of the process variation rates. Considering all sensor nodes and sensing time, for the temperature field, the maximum of this rate is  $0.0294 \text{ }^\circ\text{C/s}$ , and for the humidity field, the maximum is  $0.202 \text{ } \%/s$ . The means of these variation rates are  $0.00002 \text{ }^\circ\text{C/s}$  and  $0.00051 \text{ } \%/s$ , respectively. The humidity signal presents a larger variation rate than the temperature signal, being thus a faster process. As DECA considers the variation rate of the signal, the greater the variation rate, the higher the number of required transmissions. Thus, DECA

requires more transmissions for monitoring humidity than for temperature, an inherent feature of the adaptiveness of DECA. This explains the less significant lifetime increase (LI) when monitoring humidity, as shown in Fig. 6. In addition, when using DECA, as the variation rate decreases, nodes tend to sleep for larger time periods. For example, considering  $\delta = 2 \%$ , the means of the inactivity periods over all nodes and times are  $3.7118$  and  $1.1659 \text{ s}$  when DECA is applied for monitoring temperature and humidity, respectively; the maximum values of the inactivity periods are  $14.6594$  and  $10.4643 \text{ s}$ , respectively. These values show that DECA correctly adapts the sleeping periods of sensor nodes according to the behavior of the physical quantity being monitored.

Table 3 shows the packet delivery ratio (PDR) for different  $\delta$  when considering the monitoring of both signals. A 95 % confidence interval for the mean is also presented. PDRs above 90 % were obtained in all cases. For larger  $\delta$ , nodes sleep more, impacting more severely network connectivity. Moreover, we verified smaller values of PDR

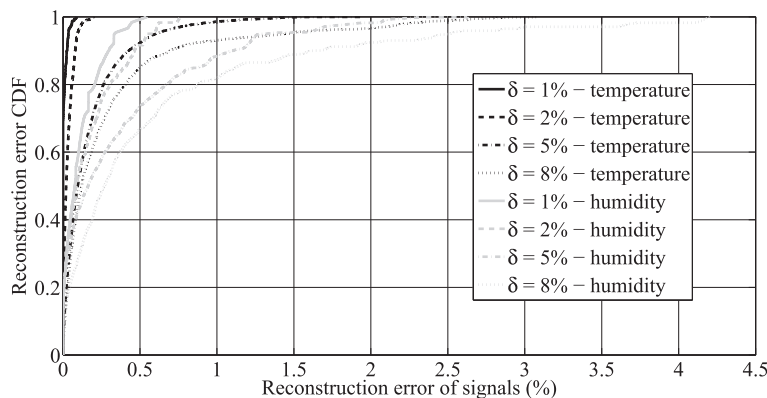


Fig. 7 Reconstruction error CDF of temperature and humidity signals

**Table 2** Maximum percentage reconstruction error (emax) × δ

δ (%)	Emax (%)—temperature	Emax (%)—humidity
1	0.1987	0.5430
2	0.2526	0.7615
5	1.9479	2.6297
8	3.0879	4.2182

for the case in which the nodes were monitoring humidity. Since this signal presents a larger variation rate, then more transmissions are required for it, which may cause more contention to access the medium and consequently collisions.

**6.2 Relay nodes**

We now analyze how DECA impacts WSN nodes that route packets for other nodes. In this scenario, the topology considered is exemplified in Fig. 4, in which node  $S_i$  has to forward packets from its neighbors to the sink node  $S_r$ . We consider cases in which  $S_i$  has 1, 5, and 13 neighbors. The WSN is monitoring temperature and δ is set as 2 %. Since γ makes nodes sleep for a fraction of the computed inactivity periods, the values in the set {0.1, 0.3, 0.5, 0.7, 0.9} are examined. Figure 8 shows the evaluation of network connectivity in two perspectives: (i) micro, by analyzing the local connectivity of the relay with the success ratio metric and (ii) macro, by analyzing the packet delivery ratio metric. In each case, the presented results are means for ten runs and the 95 % confidence intervals are also shown.

As γ increases, the node  $S_i$  sleeps for a period of time closer to the minimum inactivity period among the ones of its neighbors. It is expected to augment the probability of  $S_i$  not to forward packets from its neighbors because it may sleep for longer periods. In Fig. 8, we can see this behavior. As γ increases, there is a decrease in success ratio (SR). Furthermore, if  $S_i$  has to forward packets from a larger number of neighbors, this situation worsens. For the same reason, it is also observed that as γ increases, PDR decreases, meaning that the network is losing connectivity.

Since in this work one aims at the energy-efficient reconstruction of a monitored process by a WSN, it is important to evaluate how γ affects the energy consumption of sensor nodes. For the topology exemplified in

**Table 3** Packet delivery ratio (PDR) × δ

δ (%)	PDR (temperature)	PDR (humidity)
1	0.9774 ± 0.0045	0.9546 ± 0.0050
2	0.9750 ± 0.0045	0.9500 ± 0.0030
5	0.9420 ± 0.0050	0.9333 ± 0.0100
8	0.9180 ± 0.0150	0.9166 ± 0.0080

Fig. 4, where  $S_i$  has to forward packets from its neighbors, one knows that the IP of  $S_i$  ( $IP_i$ ) is a function of the inactivity period of its neighbors and also of γ, as presented in Eq. (30). As  $γ ∈ (0, 1)$ , Eq. (30) implies that the smaller the γ value, the shorter the  $IP_i$ . As a consequence, the smaller the γ, the lower the energy savings because the node wakes up more frequently and makes more transmissions. Figure 9 shows LI in function of γ, for δ = 2 % when 1, 5, and 13 neighboring nodes (# $SN_i$ ) are served by  $S_i$  as router. As expected, one observes that increasing γ leads to an increase of the network lifetime independently of # $SN_i$  because nodes can sleep for longer periods. When # $SN_i$  increases, the sleeping task becomes more critical as there are more packets to forward and the gain in network lifetime decreases, although these gains are still significant.

However, one should still ponder if connectivity and network lifetime are somehow related. Let  $SR_γ$ ,  $PDR_γ$ , and  $LI_γ$  be vectors containing the values of SR, PDR, and LI, respectively, for a given γ obtained with different runs. One can compute the correlation coefficient between  $SR_γ$  and  $LI_γ$ , between  $PDR_γ$  and  $LI_γ$ , and between  $PDR_γ$  and  $SR_γ$ , to evaluate the joint behavior of each variable pair. The correlations coefficients are given by

$$Corr_γ(SR, LI) = \frac{\langle SR_γ, LI_γ \rangle}{\|SR_γ\| \|LI_γ\|}, \tag{34}$$

$$Corr_γ(PDR, LI) = \frac{\langle PDR_γ, LI_γ \rangle}{\|PDR_γ\| \|LI_γ\|}, \text{ and} \tag{35}$$

$$Corr_γ(PDR, SR) = \frac{\langle PDR_γ, SR_γ \rangle}{\|PDR_γ\| \|SR_γ\|}. \tag{36}$$

In these, (x, y) is the inner product between x and y and  $\|x\|$  is the norm of x.

Figure 10 shows these correlations for DECA with δ = 2 % with different neighboring nodes. The solid and dashed curves present the correlations between the success ratio and lifetime and between packet delivery ratio and lifetime, respectively. In these curves, it is observed that the absolute value of the correlation reaches the maximum at γ = 0.5, where the statistical relationship is stronger.

The dotted curves in Fig. 10 show the correlations between the packet delivery ratio and success ratio. Since both metrics decrease in function of γ (as seen in Fig. 8), their correlation is positive.

**6.3 Network scalability**

The increase in the number of nodes in a WSN may cause more transmissions and more contention for the physical medium. This may influence energy conservation by sensor nodes and network connectivity and also the

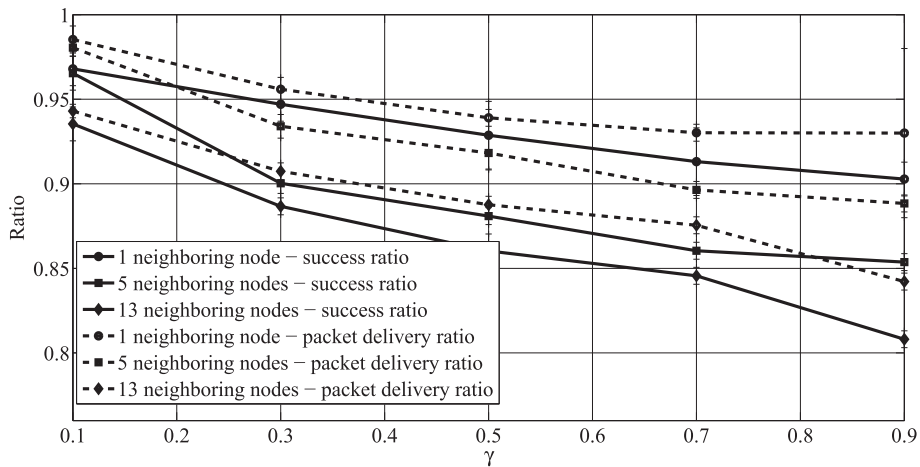


Fig. 8 Success ratio and packet delivery ratio × γ for δ = 2.0 %

reconstruction of the monitored process. Thus, we evaluate network lifetime, packet delivery ratio, and reconstruction error of the sensed field, considering WSNs having different quantities of sensors. The impact of the network scalability on DECA is evaluated considering the monitoring of temperature [21] with 15, 30, and 50 nodes; which translate into varying node density, since the monitoring area is unaltered.

Figure 11 shows LI in function of δ for the 15-, 30-, and 50-node WSNs. As expected, in the three cases, as δ increases also does LI (this was already observed in Fig. 6). One can also observe that for a given δ there is an increment in LI as the sensor density increases. Since network lifetime is defined here as the time period until the energy of the first node ends, the energy of routers may end sooner as the sensor density diminishes. If the WSN has more nodes then its sensor density increases and packet forwarding tasks can be shared among more routers, resulting in a better balance of energy consumption, redounding in the network lifetime increase.

Another aspect that can be impacted by sensor density is network connectivity. The greater the number of nodes in the monitored area, the greater the number of transmissions, which could lead to an increase in the contention for the medium. This behavior may result in more collisions, thus affecting network connectivity. In Fig. 12, this is addressed by presenting the PDR for the three different network densities, where a decrease in the PDR can be observed as the quantity of nodes (network density) increases.

Moreover, the impact of the scalability in network connectivity may affect the reconstruction of the monitored field. As packets are lost, there are fewer samples available to reconstruct the field. Corroborating this, one observes in Fig. 13 that the reconstruction error increases with the quantity of sensors in the network—the network density (Fig. 13 considers δ = 1 %). However, from the graphs in Fig. 13, it is also verified that the constraint imposed by the algorithm of keeping the reconstruction error within a prescribed fraction is satisfied, even for the 50-node WSN.

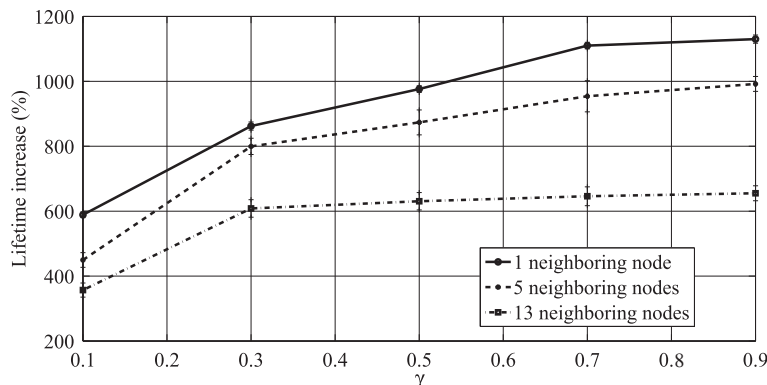
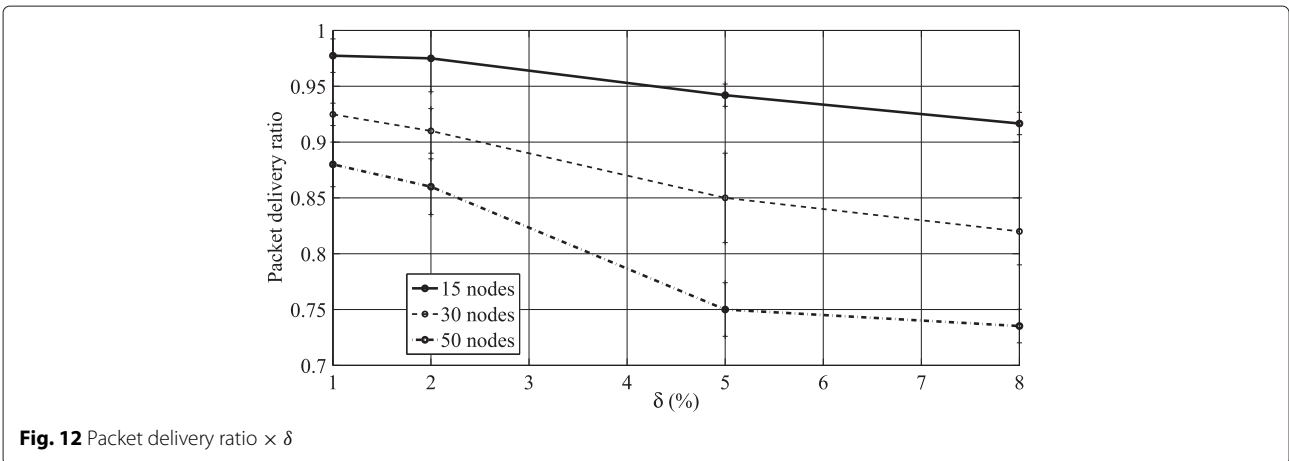
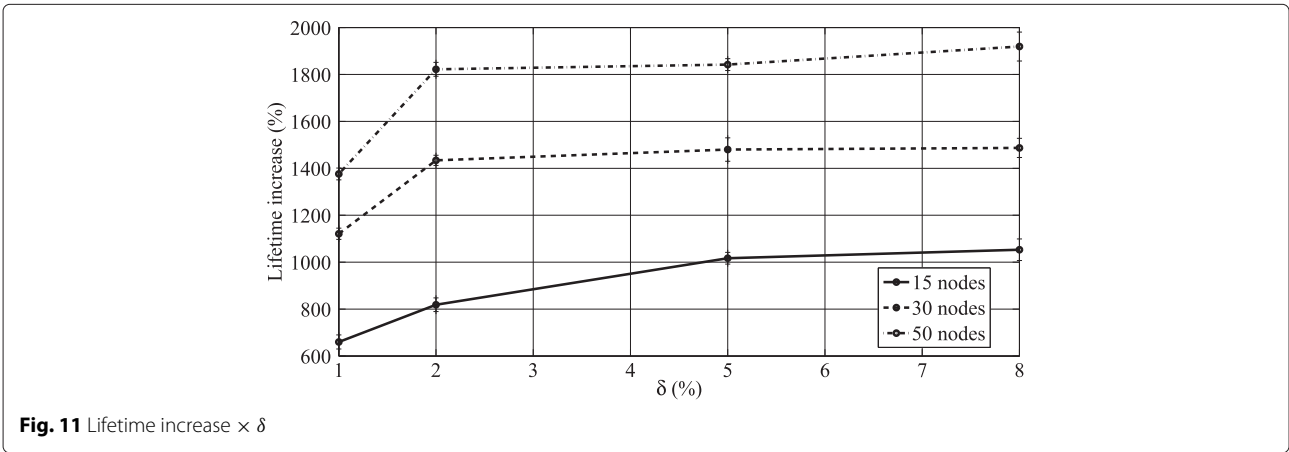
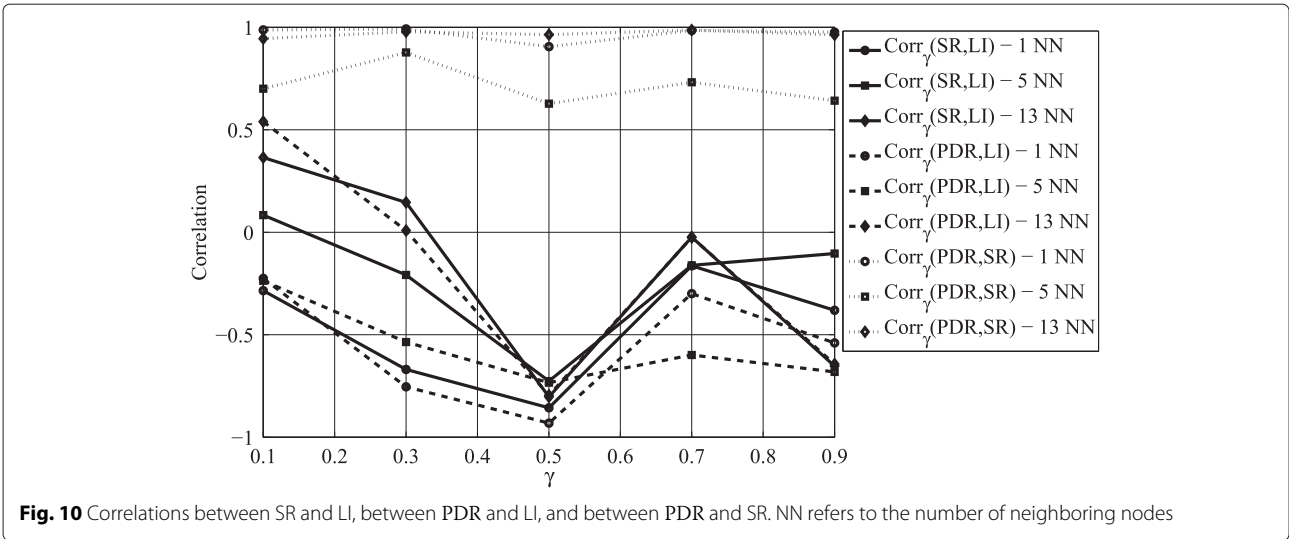


Fig. 9 Lifetime increase × γ for δ = 2.0 %



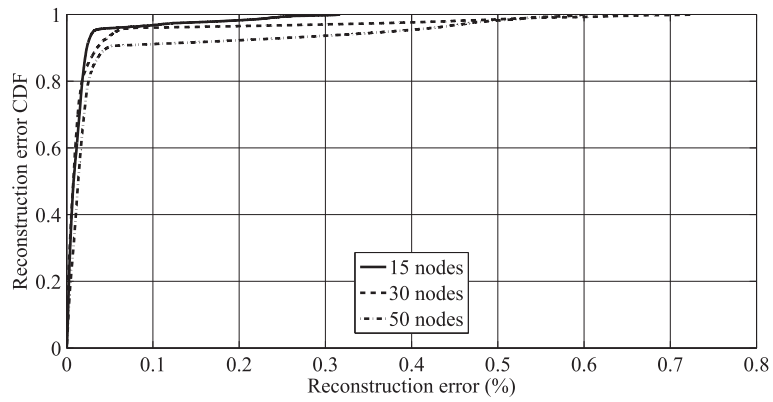


Fig. 13 Reconstruction error CDF for  $\delta = 1.0\%$

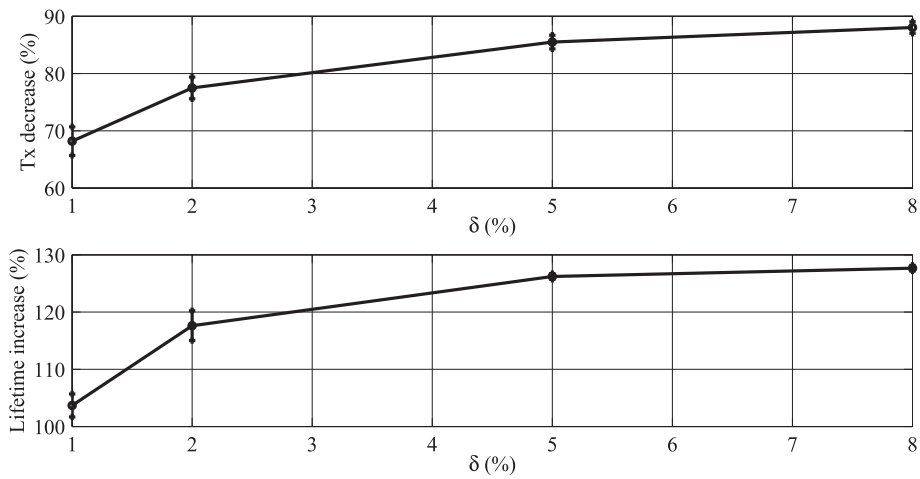


Fig. 14 Transmission decrease and lifetime increase  $\times \delta$ , for a time-base value of 1 s

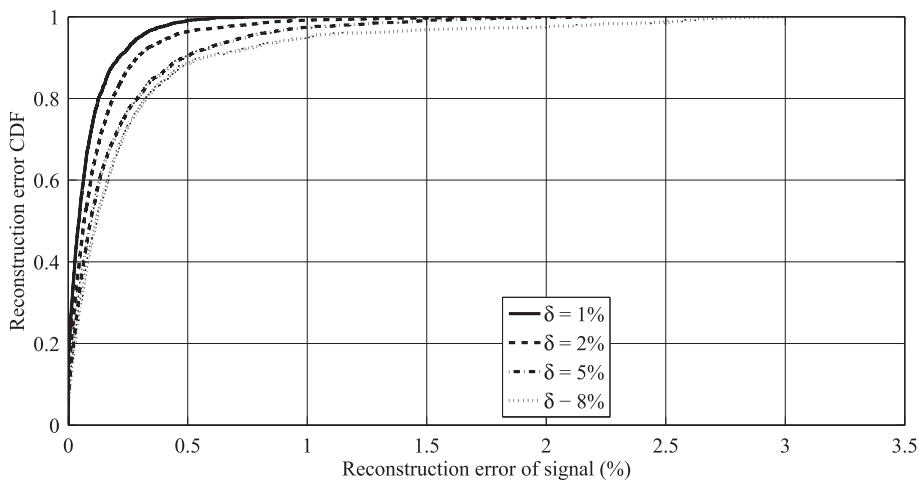


Fig. 15 Reconstruction error CDF of temperature signal, for a time-base value of 1 s



#### 6.4 Time granularity

In the previous analyses, time granularity was 0.1 s; the process is assumed to be sampled at this rate, and the nodes of the WSN that we have compared DECA against simply sample the signal of interest at 0.1 s and transmit it to the sink. In this section, we investigate the performance of DECA when other time bases are considered. The aim of such study is to evaluate the performance of DECA for processes having different speeds in function of time. We do that by using three scenarios of different signal sampling rates 0.1, 1, and 31 s. For this experiment, one reconstructs the temperature signal and a fifteen-node WSN is used. We also set  $\gamma = 0.5$  and thresholds  $\delta \in \{1\%, 2\%, 5\%, 8\%\}$  are employed.

Figure 14 shows the percentage reduction in the amount of transmissions and the network lifetime increase, in function of  $\delta$ , with respect to a WSN in which the sensors simply sample and transmit measurements to the sink, with  $T_{\text{granularity}} = 1$  s. A significant reduction in the amount of transmissions on the network and a significant lifetime increase (of more than 120 %) are observed. These are similar to the results observed for  $T_{\text{granularity}} = 0.1$  s. Albeit the transmission decrease is very similar in both cases, the gains in network lifetime are now smaller. This reduction in the gain of network lifetime is explained by the fact that the initial energy of nodes is the same (2) for the two values of  $T_{\text{granularity}}$ .

The cumulative distribution function (CDF) of the reconstruction error of the monitored temperature signal for different  $\delta$  is presented in Fig. 15. It can be observed in Fig. 15 that the reconstruction error for  $T_{\text{granularity}} = 1$  s is always smaller than the threshold, obeying the proposed constraint.

We increased the time granularity up to 31 s obtaining similar results. Table 4 shows the PDR for time-base values of 0.1, 1, and 31 s and different  $\delta$ . Ninety-five percent confidence intervals for the mean are presented. PDRs around 90 % and above are obtained, showing that the network connectivity is sustained for the different scenarios.

## 7 Conclusions

In this work, we proposed a distributed energy conservation algorithm (DECA) for WSNs, in order to increase

the network autonomy in monitoring applications. DECA uses data about the sensed process that is available at a node to estimate measured values and sleeping periods. By design, DECA imposes the reconstruction error of the monitored process to be lower than a predefined value. This is done by using linear predictors; these allow to easily find bounds on the sleeping periods for each node.

In DECA, the decision of how long a node can sleep is taken locally by each sensor node, that is, it is distributed and decentralized. To consider data routing, nodes transmit their sleeping periods together with the monitored process data so that routers can account for that in their decisions to sleep or not. A sleeping period factor is used in DECA to increase the probability of a routing node to be awake to forward packets from its neighbors. Network relay is a key challenge for energy conservation in WSN. In order to investigate if relay nodes manage their routing and sleeping tasks appropriately when running DECA, we proposed to evaluate the success ratio, which accounts for network connectivity in a micro perspective (locally). We should notice that DECA can be used together with energy-aware routing algorithms to further improve the WSN autonomy.

DECA was evaluated by simulation using actual signals. The reconstruction error, the transmission decrease, and the network lifetime were obtained. Results corroborate (i) that the error of the reconstructed process at the sink is kept within the desired fraction imposed and also (ii) that network lifetime is increased as compared to WSNs where the nodes simply transmit data periodically. The results indicate that for larger networks, the network lifetime increases more. The larger the network, the more possible the emergence of energy balancing, as there are more relays for forwarding packets from sources to the sink. In addition, we observed a decrease in both success ratio and packet delivery ratio with the increase of the sleeping factor and also with the increase of the amount of neighboring nodes. At last, we observed that as the sleeping factor increases, energy conservation also increases because nodes can sleep for larger periods of time.

#### Acknowledgements

This work has been supported by FAPERJ and CNPq.

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>Coordenação de Telecomunicações, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ), Petrópolis, Brasil. <sup>2</sup>Programa de Pós-Graduação em Engenharia Eletrônica, Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, Brasil.

Received: 4 August 2015 Accepted: 28 June 2016

Published online: 08 July 2016

**Table 4** Packet Delivery Ratio (PDR)  $\times \delta$ , for time-base values of 0.1, 1, and 31 s

$\delta$ (%)	PDR	PDR	PDR
	( $T_{\text{granularity}} = 0.1$ s)	( $T_{\text{granularity}} = 1$ s)	( $T_{\text{granularity}} = 31$ s)
1	0.9774 $\pm$ 0.0045	0.9579 $\pm$ 0.0018	0.9182 $\pm$ 0.0013
2	0.9750 $\pm$ 0.0045	0.9543 $\pm$ 0.0012	0.8959 $\pm$ 0.0017
5	0.9420 $\pm$ 0.0050	0.9341 $\pm$ 0.0020	0.8944 $\pm$ 0.0013
8	0.9180 $\pm$ 0.0150	0.9106 $\pm$ 0.0033	0.8815 $\pm$ 0.0014

## References

- IF Akyildiz WSu, Y Sankarasubramaniam, E Cayirci, Wireless sensor networks: a survey. *Comput. Netw.* **38**(4), 393U–422 (2002). doi:10.1016/S1389-1286(01)00302-4
- R Beckwith, D Teibel, P Bowmen, in *Proceedings of the 2nd International Conference on Pervasive Computing*. Pervasive computing and proactive agriculture (IEEE, Vienna Austria, 2004)
- U Raza, A Camera, AL Mutphy, T Palpanas, GP Picco, in *2012 IEEE International Conference on Pervasive Computing and Communications*. What does model-driven data acquisition really achieve in wireless sensor networks? (IEEE, Lugano, 2012), pp. 19–23. doi:10.1109/PerCom.2012.6199853
- R Szewczyk, E Osterweil, J Polastre, Habitat monitoring. *Commun. ACM.* **47**(6), 393–422 (2002). doi:10.1016/S1389-1286(01)00302-4
- ZM Wang, S Basagni, E Melachrinoudis, C Petrioli, in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. Exploiting sink mobility for maximizing sensor networks lifetime (IEEE, Hawaii, 2005), pp. 3–6. doi:10.1109/HICSS.2005.259
- G Anastasi, M Conti, MD Francesco, A Passarella, Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks.* **7**(3), 537–568 (2009). doi:10.1016/j.adhoc.2008.06.003
- PS Sausen, JRB Sousa, MA Spohn, A Perikisich, AMN Lima, in *Proceedings of the 15th Annual Meeting of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. Dynamic power management with scheduled switching modes in wireless sensor networks (IEEE, Istanbul, 2007), pp. 1–8. doi:10.1109/MASCOTS.2007.20
- IF Akyildiz, MC Vuran, OB Akan, in *Proceedings of the Modeling and Optimization in Mobile, Ad Hoc and Wireless Sensor Networks*. On exploiting spatial and temporal correlation in wireless sensor networks (IEEE, UK, 2004), pp. 71–80
- DO Cunha, RP Laufer, IM Moraes, MDD Bicudo, PB Velloso, OCMB Duarte, A bio-inspired field estimation scheme for wireless sensor networks. *Ann. Telecommun. Special Issue Wirel. Sensor Netw.* **60**(7/8), 806–818 (2005). doi:10.1007/BF03219948
- D Gao, L Zhang, H Wang, Energy saving with node sleep and power control mechanism for wireless sensor networks. *J. China Univ Posts Telecommun.* **18**(1), 49–59 (2011)
- FR Henriques, L Lovisol, MG Rubinstein, in *Proceedings of the 7th International Telecommunications Symposium*. An innovated-based algorithm for energy conservation in multihop wireless sensor networks (SBRT, Manaus Brazil, 2010)
- R Mah, A Tamhane, S Tung, A Patel, Process trending with piecewise linear smoothing. *Comput. Chem. Eng.* **19**(2), 129–137 (1995). doi:10.1016/0098-1354(94)E0042-L
- S Nguyen-Xuan, S Oh, A Sunshin, in *Advanced Communication Technology (ICTACT), 2014 16th International Conference On*. Ee-mac: Energy Efficient-Medium Access Control for Periodic Applications in Border Surveillance Wireless Sensor Networks (IEEE, Pyeongchang, 2014), pp. 693–697
- M Sha, G Hackmann, C Lu, in *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*. Energy-Efficient Low Power Listening for Wireless Sensor Networks in Noisy Environments (ACM, Philadelphia, PA, 2013), pp. 277–288
- L Tang, Y Sun, O Gurewitz, DB Johnson, in *INFOCOM, 2011 Proceedings IEEE*. Pw-mac: An Energy-Efficient Predictive-Wakeup Mac Protocol for Wireless Sensor Networks (IEEE, Shanghai, 2011), pp. 1305–1313
- R Ramanathan, R Rosales-Hain, in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment, vol. 2 (IEEE, Tel Aviv, 2000), pp. 404–413
- WR Heinzelman, A Chandrakasan, H Balakrishnan, in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference On*. Energy-Efficient Communication Protocol for Wireless Microsensor Networks (IEEE, Hawaii, 2000), p. 10
- X-C Hao, M-Q Wang, S Hou, Q-Q Gong, B Liu, Distributed topology control and channel allocation algorithm for energy efficiency in wireless sensor network: from a game perspective. *Wirel. Pers. Commun.* **80**(4), 1–21 (2014)
- Y Panthachai, P Keeratiwintakorn, in *Proceedings of the 4th International Joint Conference on Computer Science and Software Engineering*. An energy model for transmission in telos-based wireless sensor networks (Khon Kaen University, Khon Kaen, Thailand, 2007)
- J Polastre, R Szewczyk, D Culler, in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. Telos: Enabling ultra-low power wireless research (IEEE, 2005), pp. 364–369. doi:10.1109/IPSNS.2005.1440950
- labwsn IB, (2004). <http://db.csail.mit.edu/labdata/labdata.html>. Accessed 6 May 2016
- A Deshpande, C Guestrin, SR Madden, JM Hellerstein, W Hong, in *Proceedings of the Thirtieth International Conference on Very Large Data bases-Volume 30*. Model-Driven Data Acquisition in Sensor Networks (VLDB Endowment, Toronto, 2004), pp. 588–599
- MEM Campista, MG Rubinstein, *Advanced routing protocols for wireless networks*, 1st edn. (Wiley-IEEE Press, Hoboken, NJ, 2014). ISBN: 978-1-84821-627-3
- C Perkins, E Belding-Royer, S Das, in *RFC 3561*. Ad hoc on demand distance vector routing (aodv) (IETF, 2003)
- A Cervin, D Herinksson, B Lincoln, J Eker, K-E Arzén, How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime. *IEEE Control Systems Magazine.* **23**(3), 16–30 (2003). doi:10.1109/MCS.2003.1200240
- J Zheng, MJ Lee, A comprehensive performance study of iieee 802.15.4. *Sensor Netw. Oper.* IEEE Press Wiley Intersci. **Chapter 4**, 218–237 (2006)
- P Baronti, P Pillai, V Chook, S Chessa, A Gotta, YF Hu, Wireless sensor networks: a survey on the state of art and the 802.15.4 and zigbee standards. *Comput. Commun.* **30**(7), 1655–1695 (2007). doi:10.1016/j.comcom.2006.12.020

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)