ISO/IEC/IEEE 21451 Compliant Sensor Nodes for Energy-Aware Wireless Sensor Networks

G. Giorgi

University of Padua, Department of Information Engineering – DEI
via G. Gradenigo 6/b, I–35131 Padova, Italy – e-mail: giada.giorgi@unipd.it

***Abstract*—A great attention is focused nowadays on the developing of energy-aware Wireless Sensor and Actuator Network (WSAN) and in particular on implementing network control algorithms by exploiting the ability of a sensor to reliably estimate and report its energy status at network level where energy-saving strategies can be implemented. This poses the attention on how a sensor may transfer information regarding its energy resources at network level in a standardized way. In this paper we face the problem concerning the realization of energy-aware WSAN by using ISO/IEC/IEEE 21451 compliant sensor nodes. To this end, some guidelines on how exploiting the capability of the current version of the ISO/IEC/IEEE 21451 standard to obtain information about the energy-status of network nodes will be reported and, furthermore, some extensions for overcoming the existing shortcomings will finally be proposed.** [1]

I. INTRODUCTION

Developing network-level applications for Wireless Sensor and Actuator Networks (WSAN) can be greatly facilitated by the availability of devices that can be queried using a common language [1]. An important requirement for a generic sensor node is the guarantee of interoperability with other network devices, even if belonging to different manufacturers. This feature is commonly summarized by the concept of *plug-and-play*, by relying on devices that can be added, upgraded or removed from a network without affecting the whole system. Plug-and-play capability can be achieved through the definition of standardized procedures for adding new devices in a network, upgrading the existing ones and, finally, removing that no longer functioning. Furthermore, to guarantee the interconnection of WSAN to the Internet, as claimed in the emerging Internet of Things (IoT) paradigm, sensor nodes need to be addressed and connected to the user network through the Internet Protocol (IP) [2].

To this end, the family of standard ISO/IEC/IEEE 21451 [3] introduces a set of valuable features by extending the basic concept of *smart transducer*, meaning a device that provides functionality beyond those strictly necessary to generate a correct representation of the acquired or controlled quantity. A first feature involves the definition of standardized Transducer Electronic Data Sheets (TEDS) providing a comprehensive description of the transducer's characteristics. In the second instance the ISO/IEC/IEEE 21451.0 standard decouples the aspects more strictly pertaining the communication from that related instead to the sensing or control process. This is obtained by introducing two logical modules: a Network Capable Application Processor (NCAP) that supports communications toward the user network by exploiting the IP protocol and a Transducer Interface Module (TIM) which provides an interface toward sensors and actuators within a smart transducer. These two modules communicate each other through a standardized Transducer Independent Interface (TII). In WSANs the TII is based on common wireless communication protocols such as 802.11, Zigbee and 6LowPAN and is standardized in the ISO/IEC/IEEE 21451.5 document [4]. In this case the module TIM is usually called Wireless TIM (WTIM). Finally, the ISO/IEC/IEEE 21451 standard defines an Application Programming Interface (API), called *HTTP API*, by which transducer data and TEDS can be directly read from the user network by means of the HTTP (Hyper Text Transfer Protocol) protocol, simplifying in this way the development of IoT applications (see for instance [5], [6], [7]).

On the other hand, to prolong lifetime and time availability of network resources and reduce, at the same time, the economic costs related to the replacement of exhausted batteries, energy aspects need to be carefully evaluated during the development of network applications. Some architectural and algorithmic guidelines that a network designer should consider to enhance the energy awareness of a WSAN were firstly introduced in [8]. In general, different strategies can be adopted both at sensor level, as for instance in [10] where an adaptive sampling algorithm was proposed to reduce the energy consumption, and at network level, where the attention is more focused on implementing energy-aware control algorithms for minimizing the whole energy consumption in the network. In [9] for instance the lifetime of an harvesting-based WSAN is analyzed when different battery management policies are implemented. Generally, energy-saving strategies discussed in literature are based on the knowledge of the available resources in a network in a given instant. Significant improvements can therefore be achieved by exploiting the ability of a sensor to reliably estimate and report its energy status.

This poses the attention on how a sensor may transfer information regarding its energy resources at network level in a standardized way. Several solutions have already been proposed in literature for specific applications. In an industrial automation context for instance, the WirelessHART protocol [11] employs *health report* packets to convey the energy state of a node and the quality of its neighbor paths. Information conveyed by these packets are usually thought for optimizing

---

[1] This work has been supported by the Dept. of Information Engineering of the University of Padova under PRAT Contract no. CPDA112224 - Energy-autonomous Wireless Sensor Networks: from efficient sensor-level energy harvesting to intelligent network-level management.

energy resources during routing operations. On the other hand, in the home automation context, information about the state of the battery in a given device can be obtained by *command class battery* messages defined in the Z-Wave wireless protocol [12]. Another example can be found in Android-based devices where the concept of *power profiles* [13] is extensively used for monitoring the energy state by developing ad-hoc applications directly on-board of the battery-powered device. Conversely, aspects related the management of energy resources in a sensor node have not yet been specifically considered in the ISO/IEC/IEEE 21451 standard.

The purpose of this paper is twofold: to provide some guidelines on how exploiting the capability of the current version of the ISO/IEC/IEEE 21451 standard to obtain information about the energy-status of network nodes and, furthermore, to introduce some extensions to the Standard for overcoming the existing shortcomings. The paper has been organized as follows: the concept of energy map is firstly introduced in Section II and further extended in Section III. In Section IV two examples of energy monitoring system are discussed and finally tested in Section VI. TEDSs describing energy resources in a network are instead introduced in Sec. V.

II. ENERGY MAP DEFINITION

Information about the energy status of a network can be efficiently summarized by an *energy map*, namely a logic diagram containing information about the energy status of each node in the network. This diagram is usually provided by a network-level application that periodically queries the network nodes about their energy resources.

A. Network topology

The first operation for obtaining an energy-map concerns the definition of the network topology. This corresponds to a high level description of all network nodes and their logical interconnections. Since WTIM-to-WTIM communication is allowed (for routing operations for instance) as specified in the ISO/IEC/IEEE 21451.5 standard, different kinds of physical network can be implemented (mesh, point-to-point, and so on). NCAP and associated WTIMs in fact need not be able to communicate directly, that is by a single radio hop, but also by routing traffic across different network nodes. A WTIM however shall only be registered with a single NCAP, which acts as a gateway towards the user network.

The topology of a network is obtained through the discovery procedure specified by the ISO/IEC/IEEE 21451 standard. This procedure is based on the exchange of Hyper Text Transfer Protocol (HTTP) messages, as specified by the *HTTP API*, between the network application and the related NCAPs. The path field of such HTTP messages (see [14] for more information) contains a *Discovery API* command, respectively a *TIMDiscovery* command, that returns the list of all TIMs connected to a given NCAP or a *TransducerDiscovery* command that reports all transducers available in the specified TIM.

Discovery API commands are translated into a set of standardized messages at physical level for determining if a communication channel between a given NCAP and a TIM is available for data exchange operations. To keep track of the status of various network nodes, finite-state machines (FSMs) are introduced for each entity that plays a key role in the network establishment. Therefore a FSM is introduced for each communication interface (the simplest case refers to only one communication interface towards many TIMs) inside an NCAP and for each TIM that results to be registered through a given communication interface.

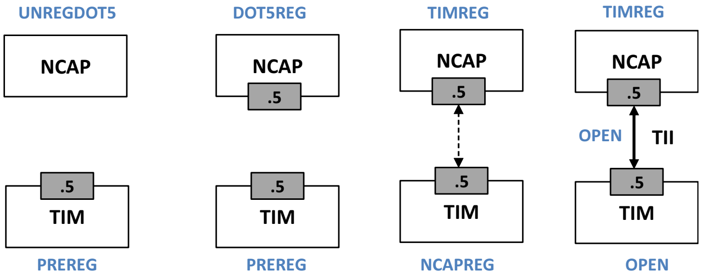

Fig. 1. Establishment of a communication channel between NCAP and TIM.

At the NCAP power on or after a reset procedure, the FSMs are initially set to the $unRegDot5$ value since no communication interface has already been associated with the NCAP (first example in Fig.1). Their status is upgraded to the $dot5Reg$ value after the registration procedure (second example of Fig. 1). At this point, one or more communication interfaces exists, but no TIM has yet been connected. After TIM registration, the status of the corresponding FSM is changed to the $timReg$ value (third example in figure). A procedure for establishing a communication channel is finally started by the NCAP by issuing an *Open Command* to the corresponding TIM. If this procedure is successful, a communication channel between TIM and NCAP is open. Consequently, the status of the corresponding FSM is upgraded to $open$ (last example in Fig. 1).

A *TIMDiscovery* command provides the list of all TIMs currently in the *open* state in a given NCAP. It is important to remark that WSANs are dynamic networks that might modify their topology over the time. This causes the changing of the status of FSMs inside an NCAP. Therefore, a *TIMDiscovery* command might return a different list of TIMs in different time instants. For keeping track of the network evolution, the discovery procedure must be periodically executed. However, this procedure is cost-demanding both in terms of bandwidth and energy consumption and must be carefully planned.

An example is provided in Fig. 2. The network consists in an NCAP and three TIMs: tim_1, which provides an interface toward sensors S_1 and S_2, tim_2, interfaced to S_3 and finally tim_3, which represents the interface for actuators A_1 and A_2. As can be noted, tim_3 results not more available.

B. Energy sources

The second operation for defining an energy map consists in specifying all the energy sources available in the network.

The 21451.0 standard allows to specify whether a network node is powered by a battery or not. This information is provided by the parameter *Battery* in the 21451.5 PHY TEDs.

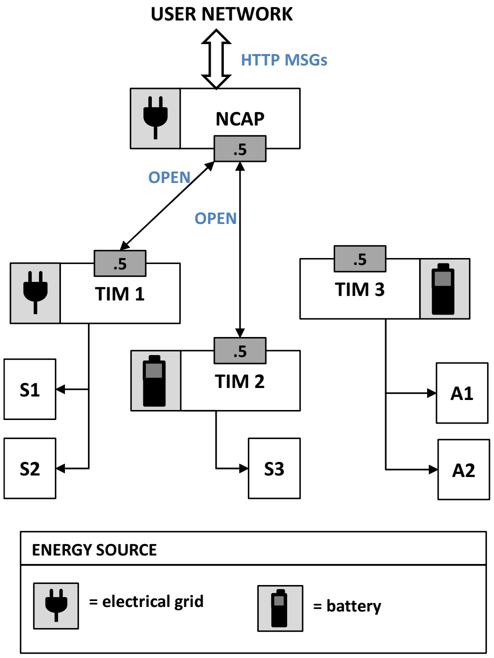

Fig. 2. Example of energy map for a WSAN based on IEEE1451 compliant sensor nodes.

A non zero value indicates a device powered by a battery, while a zero value refers to another kind of energy source. This parameter can be read through the commands specified in the *TEDS Manager API*. This API supports retrieving TEDS data associated with a specified TIM from a specified NCAP. In this case the path of the corresponding HTTP message identifies a *ReadTEDS* command.

By exploiting this knowledge, we can find (see Fig. 2) that tim_2 and tim_3 are powered by a battery, while $ncap$ and tim_1 by a different energy source, for instance the electrical grid. No information against is available for sensors and actuators, which are supposed to be fed up by the same energy source of the corresponding TIM.

C. *Energy status*

The last operation involved in the definition of a network energy map aims at monitoring the energy status of each network node. A very simple solution for solving this problem consists in querying periodically each sensor node to known its energy level and will be described in Sec. IV where other different strategies will be further discussed.

III. ENHANCED ENERGY MAP

Let we consider the same WSAN introduced in the previous Section. The new energy map illustrated in Fig. 3 brings about some remarkable improvements when compared with the previous reported in Fig. 2, in particular:

- the energy source of each device in the network - NCAP, TIMs and transducers - is specified; in Fig. 2 energy sources were specified only for NCAPs and TIMs.

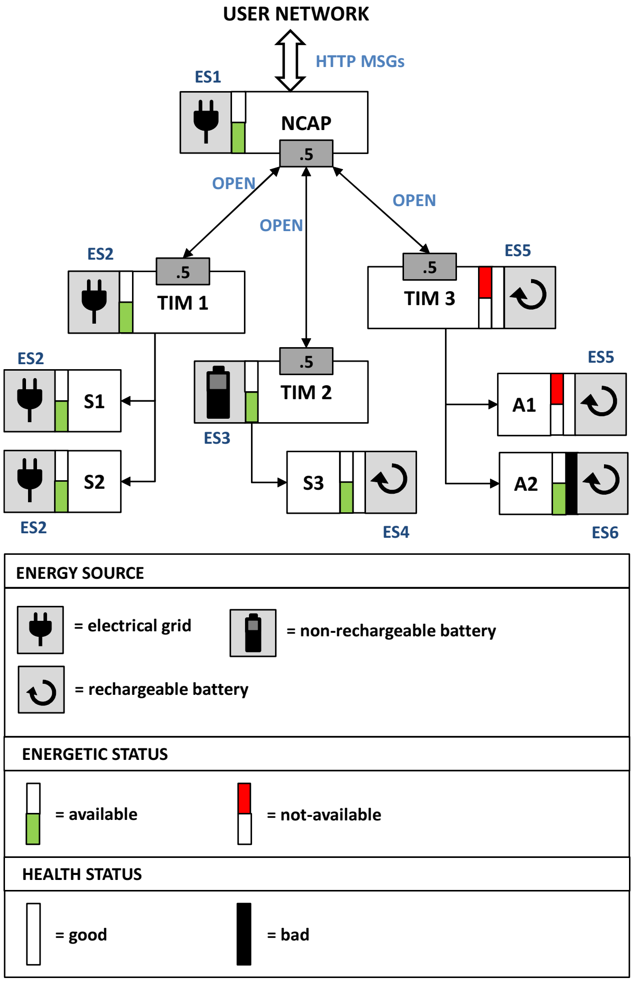

Fig. 3. Enhanced energy map for an energy-aware sensor network based on IEEE1451 compliant sensor nodes.

- energy can be drawn from the electrical grid, a non-rechargeable battery or a rechargeable battery; in the previous example the only information available concerned whether a device was powered by a battery (without any distinction about non-rechargeable batteries or rechargeable batteries instead) or not;
- the energy availability for a given energy source is known at network level, while in the previous example no information was provided.

This enhanced energy map can be obtained as briefly illustrated in the following.

A. *Nework topology*

Network topology is obtained by the discovery procedure defined in the ISO/IEC/IEEE 21451.0 standard, as previously described in Sec. II-A.

B. *Energy sources*

As can be seen in Fig. 3, six different energy sources are used, respectively: ES_1 and ES_2 corresponding to the electrical grid (different energy sources are reported since the corresponding powered devices are connected to the electrical grid in different points), ES_3 corresponding to a non-rechargeable battery and finally ES_4, ES_5 and ES_6 to rechargeable batteries. The energy map further specifies which energy source feeds a given network device. For instance, ES_1 provided energy to the $ncap$, ES_2 to tim_1 and corresponding sensors S_1 and S_2, ES_3 feeds tim_2, while S_4 is fed on by the energy source ES_4, finally tim_3 and A_1 are powered by ES_5 while the second actuator A_2 by ES_6.

These information can be efficiently summarized in a specific electronic data sheet, as introduced in Sec. V-A.

C. *Energy status*

The amount of energy that can be drawn from the electrical grid is about infinity, therefore this kind of source can be considered with good approximation always available.

The amount of energy stored in a battery instead may vary and needs to be continuously monitored. This can be accomplished by coupling information provided by two indicators, respectively the State of Charge (SoC) and the State of Health (SoH) [17]. The former provides the amount of charge stored in a given battery and is defined both for non-rechargeable batteries and for rechargeable batteries, while the second is specified only for non-rechargeable batteries and provides an estimation of how many charge-discharge cycles the battery is still able to support.

The SoC is usually expressed as the percentage of stored charge with respect to the total amount of storable charge: $SoC = 0\%$ means no charge in the battery, while $SoC = 100\%$ indicates a battery fully charged. In the same way the SoH can be expressed as percentage of the life span of a battery: $SoH = 100\%$ corresponds to a battery in optimal conditions, while a value $SoH = 0\%$ indicates a battery completely damaged.

In the example of Fig. 3, for the sake of simplicity, SoC and SoH are represented by boolean numbers that can assume only two values: 0 or 1. When a battery stores enough energy for guaranteeing the proper functionality of the powered devices $SoC = 1$, otherwise when the amount of charge in a battery falls below a given threshold $SoC = 0$. In this latter case, the energy source will be considered unavailable and the corresponding powered devices out of service.[2] Similarly, $SoH = 1$ refers to a battery in a good healthy status, while $SoH = 0$ indicates a battery in bad conditions.

An open question concerns how estimating SoC and SoH and transferring their values at network level. Possible solutions are discussed in Sec. IV.

D. *Some remarks*

For a rechargeable battery, where the charge is refilled through a renewable energy source, the energy unavailability could only refers to a temporary situation. In this case the SoC will return to a high value as soon as enough energy is scavenged. The battery needs instead to be replaced when SoH becomes low. This situation has been clearly emphasised in Fig. 3: the SoC for the rechargeable battery ES_5 is 0, therefore the corresponding powered devices tim_3 and A_1 are temporarily unavailable. The communication channel between $ncap$ and tim_3 is however in the *open* state (while in Fig. 2 tim_3 was removed from the network) even if tim_3 cannot support any communication at the moment. This avoids a new registration of tim_3 with the NCAP and the successive opening of a dedicated communication channel. The rechargeable battery ES_6 instead is in bad conditions ($SoH = 0$) even if some energy is currently available. This means that the battery cannot be recharged after that its charge is consumed and needs to be replaced as soon as possible.

It is important to remark that there are several factors in a WSAN that might cause a node to be removed from a network. Bandwidth unavailability, failures on the communication interface and unavailability of enough energy for assuring a proper functioning of the sensor node are some of such causes. Some failure conditions are unpredictable for their nature, such as the occasional unavailability of bandwidth due to external interferences or damages on network devices caused by electrostatic discharges, thermal or mechanical shocks, radiation and so on.

At this purpose, some safety criteria need to be implemented to prevent stalemate situations, which might occur when a node, not more correctly functioning, is still considered available in the network. These criteria are typically based on the behaviour of the communication channel, as proposed in [18] or [19]. At this purpose, the ISO/IEC/IEEE 21451 standards provide some methods for determining whether a node in a network is still operating or not. In particular, if an NCAP does not receive an acknowledgement from a given TIM within a specified amount of time, the corresponding TIM will be removed from the network. Generally, a given number of failed communication attempts can be tolerated before disconnecting a node from the network. The maximum number of admissible retries at physical level is specified by the parameter *Number of maximum retries before disconnect* in the PHY TEDS for wireless interfaces [4].

IV. ENERGY MONITORING SYSTEM

Reliably estimating the State of Charge and State of Health of a battery represents still nowadays a very challenging problem [15], [16]. The purpose of this work however does not concern the estimation of these quantities. The problem that we are facing is to provide solutions for making these values known at network level in a standardized manner. By exploiting the features of the Standard IEEE 1451, two different solutions can be found: a *Transducer-based solution* and a *TEDS-based solution*. Also hybrid solutions may be used, however they have not been described in this paper.

A. *Transducer-based solution*

Different solutions have been proposed in literature for estimating the SoC and SoH of a battery by onboard measurement systems [15]. The simplest solution consists, in this case, in considering the onboard measurement systems like ISO/IEC/IEEE 21451 compliant sensors, interfaced through dedicated transducer channels to a given TIM. The measured

---

[2] In general, it is possible to introduce a safety margin that allows powered devices to still work for a given amount of time, during which planning the battery replacement, after that SoC changes to a low value.

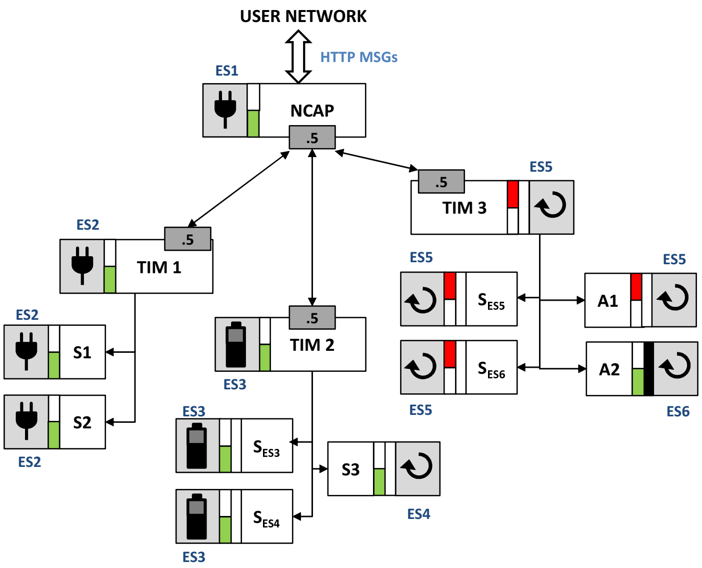

Fig. 4. Enhanced energy map for an energy-aware sensor network based on IEEE1451 compliant sensor nodes.

quantity, corresponding to the SoC or SoH, can be read from a network application in a standardized manner through the *Transducer Access API* commands.

The network topology needs to be modified however to reflect these changes, as reported in Fig. 4. In the proposed example, S_{ES_i} provides information about the battery ES_i, with $i = 3, 4, 5$ and 6, even if this sensor may be powered by a different energy source. For instance S_{ES_4} takes into account the behaviour of ES_4 but it is fed up by ES_3. It is important to remark that the estimation process and the corresponding data exchange consume energy resources, therefore the status of the battery that feeds S_{ES_i} might be significantly perturbed. This solution is suitable for systems characterized by high-capacity batteries, where information concerning the battery status can be periodically acquire and transmit to a network-level application without affecting the whole amount of energy stored in the battery.

B. *TEDS-based solution*

When network devices are powered by very low-capacity batteries, for instance in [20], the previous solution becomes infeasible. In this case it will be necessary to implement estimation algorithms directly at network level where all information necessary, such as the total capacity of the battery, the energy required for data transmission or reception, the charging rate of the battery and so on, must be known. These information are summarized in electronic data sheets, as described in Sec. V-B and V-C, and read through the *TEDS Manager API*.

It is important to note that reading operations on a given TEDS require a data exchange with the corresponding TIM only the first time. Successive operations in fact will read the desired TEDS block from the cache on the corresponding NCAP, resulting in a very efficient implementation.

Value	Meaning
0	The energy source is located within the NCAP module.
1	The energy source is located within the TIM module.
2	The energy source is on-board of a given transducer module.

TABLE II. LOCATION OF THE ENERGY SOURCE

Value	Meaning
0	The energy source is provided by the electrical grid or a centralized energy source (for instance in planes, ships, cars).
1	The energy source is drawn from a non-rechargeable battery.
2	The energy source is provided by a rechargeable battery.

TABLE III. ENERGY SOURCE

Value	Meaning
0	The powered device corresponds to the NCAP.
1	The powered device corresponds to the TIM.
2	The powered device corresponds to a sensor.
3	The powered device corresponds to an actuator.
4	The powered device corresponds to an event sensor.

TABLE IV. POWERED DEVICE

V. TEDS FOR ENERGY-AWARE NODES

In this Section specific TEDSs are proposed for devices operating in energy-aware WSAN. The first of these TEDS provides a description of the energy sources available in the corresponding network device, the second contains a battery description and the latter specifies the amount of energy consumed by the different tasks executed in a network node.

A. *Energy Source META TEDS*

The energy provisioning system for a given network device is described in the *Energy Source META TEDS*, whose structure has been reported in Tab. I. This TEDS aims at identifying all the energy sources in a given NCAP or TIM and the related powered devices.

The first field, i.e. *EnergySourceNum*, specifies the number of energy sources available within the module. The maximum number of energy sources is specified by the state variable $MaxNumES$. An identification number is then assigned to each energy source and the related information about location, kind and number of powered devices provided. Afterwards, the TEDS reports a brief description for each device powered by a given energy source. This description consists essentially in specifying the kind of powered device and in providing its identification number. The maximum number of devices that a given energy source can feed is specified by the state variable $MaxNumPd$. More details can be found in Tabs. I, II, III and IV.

B. *Battery TEDS*

The *Battery TEDS* contains all the information related to a given battery and simply reports parameters usually specified in a battery data sheet, as illustrated in Tab. V. The first field specifies the kind of battery (see Tab. VI), while the second corresponds to a boolean variable that specifies if the battery can be recharged or not. The last three parameters in the data sheet instead contain all the necessary information for estimating - even though in a very approximative manner - the State of Charge and/or State of Health of the battery at network level.

Type	Name	Description
1	EnergySourceNum	Number of energy sources in the corresponding module. Value: from 1 to $MaxNumES$.
		Energy Source Information Sub-block
2	EnergySourceId	Energy source identification number. Value: from 1 to 255.
3	EnergySourceLocation	Location of the energy source. Values are provided in Tab. II.
4	Energysource	Kind of energy source used to power the devices. Values are described in Tab. III.
5	PoweredDeviceNum	Number of powered devices from the given energy source. Value: from 1 to $MaxNumPd$.
		Powered Device Information Sub-block
6	PoweredDeviced	Kind of powered device. Values are described in Tab. IV.
7	PoweredDeviceId	Powered device identification number.

TABLE I. ENERGY SOURCE META TEDS

Type	Name	Description
1	Battery	Describes the kind of battery. Values are described in Tab. VI
2	Rechargeable	A non zero value indicates a rechargeable battery.
3	Capacity	Provides the total amount of charge in a battery measured in [A· hours].
4	RechargeCycle	Number of charge-discharge cycles that a battery can support (only for rechargeable batteries).
5	RechargeRate	Time rate for refilling the battery (only for rechargeable batteries).

TABLE V. BATTERY TEDS

Type	Name	Description
1	TIMSleep	A non negative number provides the amount of energy per time unit consumed during the sleep period of a TIM. A negative number corresponds to an unknown quantity.
2	TIMSleepTime	A non negative value corresponds to the total amount of time elapsed in the sleep state.
3	TIMActive	A non negative number provides the amount of energy per time unit consumed during the active period of a TIM.
4	TIMActiveTime	A non negative value corresponds to the total amount of time elapsed in the active state.
5	TIMActiveTaskNumber	A non zero value corresponds to the number of sub-tasks for which the corresponding energy consumption is specified.
		TIM Active Task Information Sub-block
6	TaskId	Task identification number. Possible values are reported in Tab. IX
7	TChId	Identification number of the transducer channel involved in sampling operations. It is defined only for *TaskId*=1 or 2.
8	EnergyUnit	Measurement unit for the energy, as specified in Tab. VIII
9	TaskEnergy	Energy consumed by the task.

TABLE VII. ENERGY CONSUMPTION TEDS

Value	Meaning
0	Lithium-Ion battery.
1	Lead-acid battery.
2	Nichel-cadmium battery.
3	Alkaline.

TABLE VI. BATTERY

Value	Meaning
0	energy.
1	energy per time unit.
2	energy per byte.

TABLE VIII. ENERGY MEASUREMENT UNIT

C. Energy Consumption TEDS

The *Energy consumption TEDS* describes the more significant tasks executed by a TIM from an energy-consumption point of view. To this end, the operating cycle of a TIM is modelled by a finite state machine, as proposed in [3], having two main states: a *sleep state* during which the energy consumption is usually very low and sometimes negligible and an *active state* where the node performs all the designed activities.

Value	Meaning
0	Wake-up.
1	Sample sensing for a Sensor.
2	Sample processing for an Actuator.
3	Data transmission.
4	Data reception.
5-255	User defined.

TABLE IX. TASK IDENTIFICATION NUMBER

This representation is adopted also in the corresponding TEDS, reported in Tab.VII. Usually the energy consumed during the *sleep state* is constant and can be provided by the parameter *TIMSleep* per time unit. The time elapsed in this state, reported in *TIMSleepTime*, is usually deterministic and knowable a-priori. Otherwise, it must be determined by the network application and successively used for calculating the total amount of energy absorbed.

The time spent by the TIM in the *active state* is unlikely deterministic, therefore the parameters *TIMActive* and *TIMActiveTime* are usually not determinable. For this reason the activity period is commonly split into more tasks: each task is described in a specific entry of the *TIM Active Task Information Sub-block*. For each task the TEDS provides a brief description concerning the operation done, as reported in Tab. IX, and eventually the transducer channel involved in such operation. Afterwards, the energy consumed from the task is reported. This value is normalized with respect to seconds or bytes; this latter to simplify the calculation of energy absorbed during transmission or reception of messages. It is important to note that the definition of this TEDS might require a preliminary characterization of the network device as reported both in [13] and [21]. Only the tasks that might significantly affect energy resources need however to be reported.

VI. EXAMPLES

We consider a node WTIM, powered by a rechargeable Lithium-Ion battery, which acquires data samples every $T_{sample} = 0.5$ s by an 8 bit Analog to Digital Converter, stores them in a data buffer having maximum capacity $C_{buffer} =$ 10 Kbyte and finally transmits the data currently available in the buffer to the NCAP in response to a read data-set command. All these information are summarized in the corresponding *Transducer Channel TEDS*.

All ISO/IEC/IEEE 21451 commands are coded in 16 byte-length messages, while the response message is composed by a fixed packet header of 16 bytes followed by a variable payload, which length depends on the quantity of data to send (2 for a battery measure and N_{sample} for a generic data set).

The battery discharge process is simulated by considering typical values for wireless devices based on the IEEE 802.15.4 protocol (see [22]). Under this assumption, we have chosen a constant discharge current $I_{discharge} = 20$ mA both for receiving and transmitting data through the wireless interface. Furthermore, for a data rate $R = 250$ Kbit/s, the time spent for sending or receiving n bytes results to be:

$$\Delta T(n) = 1 + \frac{8 \cdot n}{250} \quad ms \tag{1}$$

as reported in [22].

The battery has a nominal capacity $C_{battery} = 50$ μAh in accordance with the values reported in the data sheet of the rechargeable Lithium-Ion battery CBC EnerChip 5300 [23] and it employs about 24 hours to completely recharge itself by the renewable energy source. Finally, to prolong the lifetime of the battery, the sensor node is put in a sleep state when the state of charge of the battery falls below a given safety threshold, that has been set to 70%.

A. First example

In this first example no energy management system is implemented. Therefore, when enough energy is stored in the battery, the WTIM continuously acquires measurement samples, stores them in the acquisition buffer and finally sends them to the NCAP in correspondence of a read command, which will be periodically issued by the NCAP. By assuming a time period $T_{polling} = 60$ s for the polling cycle, the response message sent by the WTIM will contain almost $N_{sample} = T_{polling}/T_{sample} = 120$ bytes, as illustrated in the bottom plot of Fig. 5. Since the maximum size of a IEEE 802.15.4 packet is 128 bytes, only a single physical packet will be sent to the NCAP.

As can be noted from the top plot reported in Fig. 5, the battery SoC ($Q_{battery}$) periodically falls below the safety threshold causing the node unexpectedly enters in the sleep state and consequently be disconnected to the NCAP. In the sleep state all operations are suspended resulting in a loss of measurement data. The WTIM will be reconnected only by a discovery procedure, that is initiated from the NCAP with a time period $T_{discovery} = 1$ hour. This behavior can be observed in the middle plot of Fig. 5 where the status of the WTIM module becomes alternately *REG*, when the connection toward the NCAP is open and *UNREG* when the WTIM is removed from the network.

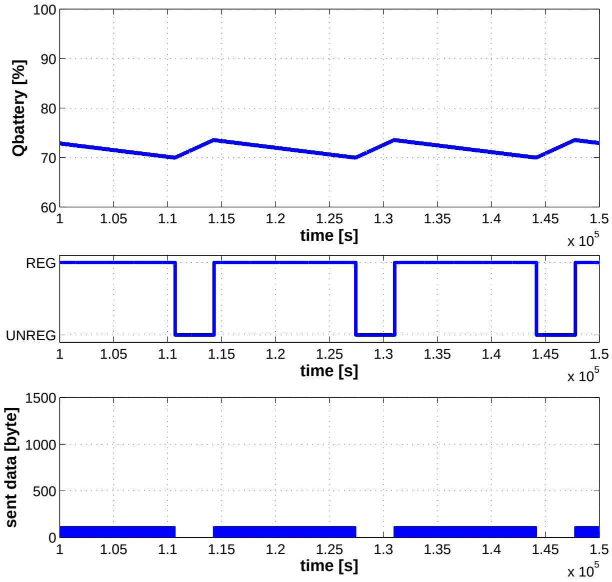

Fig. 5. Analysis of the performances achievable by an energy-unaware system.

B. Second example

The second example implements a *Transducer-based* energy monitoring system. This solution is thought for unpredictable renewable energy sources, since information about the energy resources of a given node are obtained by by directly querying the on-board battery measurement system.

The NCAP module starts by a read operation of the battery state before issuing a read data-set command to the WTIM. Only if the resulting value is greater than a given threshold (75% in the example) the NCAP will go ahead by reading the measurement data, conversely it will suspend any operation concerning the WTIM for a given amount of time. Performances achievable by this strategy are reported in Fig. 6. In this case no measurement data is lost, however data are sent in blocks of about 1200 samples, thus requiring the transmission of about 10 packets for each data set. It can be straightforwardly notice that this solution might become prohibitive when interference and other impairments affect the behavior of the physical channel.

C. Third example

This last example refers to a *TEDS-based* energy monitoring system. In this case all information related to battery and energy consumption of the sensor node are summarized in TEDSs (reported in Tab. X and Tab. XI) that will be read each time a WTIM is connected by a discovery procedure to the NCAP. TEDS parameters are used for calculating a suitable polling period $T_{polling}$ in order to satisfy the inequality:

$$Q_{charge} \geq Q_{discharge} \tag{2}$$

where $Q_{charge} = q_{recharge} \cdot T_{polling}$ is the amount of charge stored in the battery during the given time period and $Q_{discharge}$ is the charge consumed during the WTIM activity

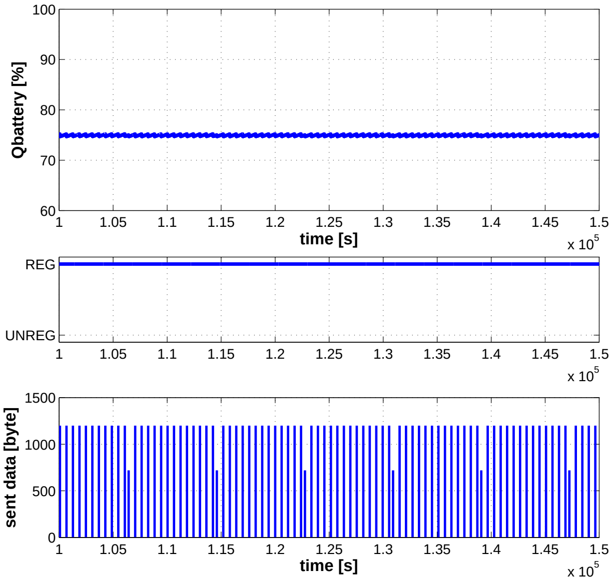

Fig. 6. Analysis of the performances achievable when a Transducer-based energy monitoring system is implemented.

in the same time period[3]. In the following we will consider, without loss of generality, only receiving and transmitting operations, thus obtaining:

$$Q_{discharge} = \frac{I_{discharge}}{C_{battery}} \cdot \left[1 + \frac{8 \cdot N_{receive}}{250}\right] + \frac{I_{discharge}}{C_{battery}} \cdot \left[1 + \frac{8 \cdot N_{transmit}}{250}\right] \quad (3)$$

where $N_{receive} = 16$ bytes is the byte length of a read command, while $N_{transmit} = 16 + N_{sample}$ is the number of bytes sent to the NCAP in a read command response. By solving this inequality we find two conditions that need to be satisfied at the same time:

$$T_{sample} \geq \frac{k_1 \cdot I_{discharge}}{C_{battery} \cdot q_{recharge}} \quad (4)$$

$$T_{polling} \geq \frac{k_2}{\frac{q_{recharge} \cdot C_{battery}}{I_{discharge}} - \frac{k_1}{T_{sample}}} \quad (5)$$

where $k_1 = 32 \cdot 10^{-6}$ and $k_2 = 3 \cdot 10^{-3}$ are numeric constants. By considering the values previously introduced, the first inequality results to be verified by assuming a sampling period $T_{sample} \geq 360$ ms. By choosing $T_{sample} = 0.5$ s, the second inequality holds true for $T_{polling} \geq 120$ s. Results achieved with $T_{polling} = 120$ s have been reported in Fig. 7. It is interesting to note that no data is lost and the length of the payload (about 240 samples) results reduced with respect to the previous example. In this case only two 802.15.4 packets will be sent to the NCAP.

This strategy is based on the assumption that the renewable energy source is always available, thus providing a constant recharge of the battery. Furthermore, the corresponding TEDSs need to be read by the NCAP each time a WTIM is connected or reconnected through a discovery procedure. However this does not result in an excessive overload, in fact by assuming 2 bytes for each TEDS entry, the *Battery TEDS* will take about 80 bytes and the *Energy consumption TEDS* in Tab. XI about 176 bytes.

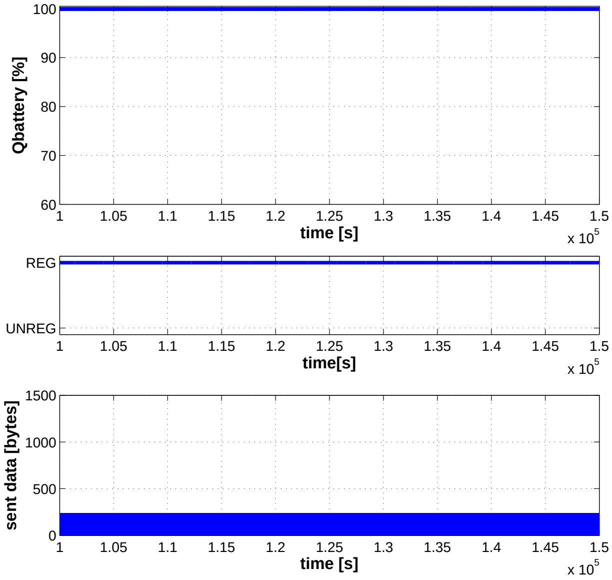

Fig. 7. Analysis of the performances achievable when a TEDS-based energy monitoring system is implemented.

TEDS parameter	Value
Battery	0 (Lithium-Ion)
Rechargeable	1 (rechargeable)
Capacity	$C_{battery}$
RechargeCycle	10000
RechargeRate	$q_{recharge}$

TABLE X. EXAMPLE OF BATTERY TEDS.

VII. CONCLUSIONS

The definition of standardized procedures for transferring information about the energy status of a network node at network level is still nowadays an open question and results to be a very challenging problem.

In this paper some valuable ideas, based on exploiting the standard ISO/IEC/IEEE 21451, have been proposed. The main

TEDS parameter	Value
TIMSleep	0
TIMSleepTime	0
TIMActive	-1 (unknown quantity)
TIMActiveTime	-1 (unknown quantity)
TIMActiveTaskNumber	2
TaskId	3 (data transmission)
EnergyUnit	1 (energy per time unit)
TaskEnergy	$I_{discharge}/C_{battery}$
TaskId	4 (data reception)
EnergyUnit	1 (energy per time unit)
TaskEnergy	$I_{discharge}/C_{battery}$

TABLE XI. EXAMPLE OF ENERGY CONSUMPTION TEDS.

---

[3] These values are supposed to be normalized with respect to the battery capacity $C_{battery}$.

goal of the proposed work is solving the underlined problem by introducing as less changes as possible on the Standard. To this end, the Application Programming Interface is kept unchanged, while customized electronic data sheets have been introduced to describe the energy provisioning scheme, battery parameters and the energy consumption within a network device.

REFERENCES

[1] E.Y. Song, K. Lee, *Understanding IEEE 1451 - Networked Smart Transducer Interface Standard*, Proc. of IEEE Instrumentation and Measurement Magazine, vol. 11, no. 2, April 2008, pp.11-17.

[2] A.P. Castellani, N. Bui, P. Casari, M. Rossi, Z. Shelby, M. Zorzi, *Architecture and protocols for the internet of things: a case study*, Proc. of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM), pp. 678-683, April 2010.

[3] IEEE Std. 1451.0 - 2007, *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats*, March 2007, pp.1-323.

[4] IEEE Std. 1451.5 - 2007, *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, October 2007, pp.1-246.

[5] E.Y. Song, K.B. Lee, *STWS: A Unified Web Service for IEEE 1451 Smart Transducers* IEEE Trans. on Instr. and Meas., vol.57, no.8, pp. 1749-1756, Aug. 2008.

[6] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, *A low-cost Internet-enabled smart sensor*, Proc. of IEEE Sensors, vol.2, pp.1549-1554, 2002.

[7] Q. Chi, H. Yan, C. Zhang, Z. Pang, L. Daxu, *A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment*, IEEE Trans. on Ind. Inform., vol. 10, no. 2, pp. 1417-1425, May 2014.

[8] V. Raghunathan, C. Schurgers, P. Sung, M.B. Srivastava, *Energy-aware wireless microsensor networks*, IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 40-50, 2002.

[9] N. Michelusi, L. Badia, R. Carli, L. Corradini, M. Zorzi, *Energy Management Policies for Harvesting-Based Wireless Sensor Devices with Battery Degradation*, IEEE Trans. on Communications, vol.61, no.12, pp.4934-4947, December 2013.

[10] C. Alippi, G. Anastasi, M. di Francesco, M. Roveri, *Energy Management in Wireless Sensor Networks with Energy-hungry Sensors*, IEEE Instrumentation and Measurement Magazine, Vol. 12, N. 2, April 2009, pp. 16-23.

[11] IEC 62591, *Industrial communication networks - Wireless communication network and communication profile - WirelessHART*, International Electrotechnical Commission, 2010.

[12] Z-Wave Alliance, *Z-Wave*, http://www.z-wavealliance.org/.

[13] Android Open Source Project,*Power Profile for Android*, https://source.android.com/devices/tech/power.html.

[14] RFC 2616, Hypertext Transfer Protocol - HTTP/1.1, June 1999.

[15] V. Pop, H.J. Bergveld, P.H.L. Notten, P.P.L. Regtien, *State of the art of battery state of charge determination*, IOP Publighing, Meas. Sci. Technol. vol. 16, R93-R110, 2005.

[16] M. Gholizadeh, F.R. Salmasi, *Estimation of State of Charge, Unknown Nonlinearities, and State of Health of a Lithium-Ion Battery Based on a Comprehensive Unobservable Model*, IEEE Transactions on Industrial Electronics, vol. 61, no. 3, pp. 1335-1344, 2014.

[17] K.S. Nga, C.S. Mooa, Y.P. Chen, Y.C. Hsiehc, *Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries*, Applied Energy, Elsevier, vol. 86, no. 9, pp. 1506-1511, Sept. 2009.

[18] H.C. Shih, J. H. Ho,B. Y. Liao, J. S. Pan, *Fault Node Recovery Algorithm for a Wireless Sensor Network*, IEEE Sensors Journal, vol.13, no.7, pp.2683-2689, July 2013.

[19] R.N. Duche, N.P. Sarwade, *Sensor Node Failure Detection Based on Round Trip Delay and Paths in WSNs*, IEEE Sensors Journal, vol.14, no.2, pp.455-464, Feb. 2014.

[20] B. Cook, R. Vyas, S. Kim, T. Thai, T. Le, A. Traille, H. Aubert, M. Tentzeris, *RFID-Based Sensors for Zero-Power Autonomous Wireless Sensor Networks*, IEEE Sensors Journal, January 2014.

[21] G. Giorgi, A. Veronese, L. Corradini, *A Method for Estimating State of Charge in Energy-Aware Wireless SEnsor Networks*, Proc. of the 19-th Symposium IMEKO TC 4, July 18-19, 2013, Barcelona, Spain.

[22] E. Casilari, J.M. Cano Garcia, G. Campos Garrido, *Modeling of Current Consumption in 802.15.4-Zigbee Sensor Motes*, Sensors, vol.10, no.8, Jun. 2010, pp. 5443-5468.

[23] Cymbet Corporation, *EnerChip EH solar energy harvester evaluation kit*, CBC-EVAL-08, www.cymbet.com.