An Energy-Efficient Architecture for the Internet of Things (IoT)

Navroop Kaur and Sandeep K. Sood

***Abstract*—Internet of things (IoT) is a smart technology that connects anything anywhere at any time. Such ubiquitous nature of IoT is responsible for draining out energy from its resources. Therefore, the energy efficiency of IoT resources has emerged as a major research issue. In this paper, an energy-efficient architecture for IoT has been proposed, which consists of three layers, namely, sensing and control, information processing, and presentation. The architectural design allows the system to predict the sleep interval of sensors based upon their remaining battery level, their previous usage history, and quality of information required for a particular application. The predicted value can be used to boost the utilization of cloud resources by reprovisioning the allocated resources when the corresponding sensory nodes are in sleep mode. This mechanism allows the energy-efficient utilization of all the IoT resources. The experimental results show a significant amount of energy saving in the case of sensor nodes and improved resource utilization of cloud resources.**

***Index Terms*—Cloud computing, energy efficiency, Internet of things (IoT), sensors, sleep interval.**

I. Introduction

WITH the advent of a new era in computation, Internet of things (IoT) [1] has emerged as a building block of ubiquitous computing [2]. IoT is a smart technology that interconnects each and every "thing" through a network in one form or another. The term "thing" includes sensors, actuators, hardware, software, and storage spread over multiple disciplines such as healthcare, industry, transport, and home appliances. The main objective of IoT is to maximize the communication of hardware objects with the physical world and to convert the data harvested by these objects into useful information without any human aid. IoT consists of three elements: hardware, middleware, and presentation. The hardware element is comprised of battery-powered embedded sensors, actuators, and communication systems. The sensors collect data from the monitoring area, and their communication hardware sends the collected data to the middleware element. An enormous amount of data received by middleware is processed and analyzed by using various data analysis tools to extract interpretable information. The presentation element of IoT is responsible for the visualization of processed data and results in a novel and easily readable form. It also receives user queries and passes them to the middleware element for necessary actions. Fig. 1 shows the elements and data transfer in IoT systems.

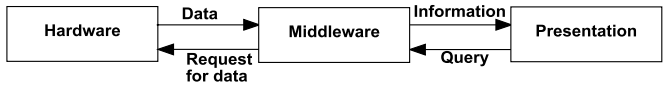

Fig. 1. Elements of IoT.

The limited battery power of hardware elements is consumed while collecting and transmitting data. The more are the data collected and analyzed, the more is the accuracy of extracted information but, at the same time, the more is the energy consumed. Due to energy limitations, there is a need to maintain a tradeoff between quality of information extracted and energy consumption by IoT systems. Moreover, the lifetime of any resource in IoT depends upon the availability of energy. The loss of energy affects the whole environment under observation. Thus, there is a prominent need to reduce energy consumption for the prolonged lifetime of resources and the effective operation of IoT systems.

Derived by the given motivations, this paper proposes a hierarchical architecture to improve the energy efficiency of IoT. The proposed architecture (PA) exploits the fact that IoT resources consume negligible energy in sleep mode. Hence, the architectural design allows the sensors to switch to sleep mode under three scenarios: first, when it is not necessary to sense the target environment in a given period of time; second, when the coverage area can be compromised for battery life; and third, when the battery level is critically low. In addition, when the sensors are in sleep mode, the PA allows the allocated middleware resources to either switch to sleep mode or get released and reprovisioned later for energy efficiency. Thus, the resources of hardware and middleware elements of IoT have been "tuned" together in the PA for better performance and energy saving.

In light of the above, the main objectives of the PA can be listed as follows: 1) to provide a mechanism for the efficient energy utilization of both hardware and middleware elements of IoT systems; 2) to predict and control the sleep interval of sensors depending upon their previous usage history and remaining battery level; and 3) to reprovision the allocated cloud resources when corresponding sensors are in sleep mode.

The rest of this paper is organized as follows. Section II investigates the work related to architecture and energy-saving

Manuscript received October 13, 2014; revised April 28, 2015, July 7, 2015, and August 14, 2015; accepted August 14, 2015.

The authors are with the Department of Computer Science and Engineering, Guru Nanak Dev University Regional Campus, Gurdaspur 143521, India (e-mail: navonline@yahoo.co.in; san1198@gmail.com).

Digital Object Identifier 10.1109/JSYST.2015.2469676

techniques used in IoT and wireless sensor networks (WSNs). Section III proposes an energy-efficient architecture for IoT, and Section IV presents its theoretical analysis. Section V provides experimental results. Section VI gives the comparative analysis of the PA with other techniques listed in Section II. Finally, Section VII concludes this paper.

II. Related Work

In 2013, Gubbi *et al.* [1] provided a clear vision of IoT and listed "energy-efficient sensing" as one of the research challenges. They presented a cloud-centric architecture of IoT and emphasized that it is applicable in various areas such as industry, home, medical systems, and many more. Later, many authors worked toward the integrated application of IoT and cloud computing in industries such as manufacturing [3], environment monitoring [4], real-time locating systems [5], energy saving [6], cloud manufacturing [7], [8], and supply chains [9]. Xu *et al.* in [10] presented a survey for the application of IoT in industries. IoT has also been used in various other applications such as those listed in [11]–[18].

Since energy efficiency is a challenging issue in IoT, many authors worked toward this direction. In 2014, Akgul and Canberk [19] proposed the concept of "Self-Organized Things" (SoT), in which the sensors undergo automatic configuration, optimization, and healing mechanisms to save energy. They explained that the sensors can be put in sleep mode when the coverage area can be compromised for energy saving. In 2014, Zhou *et al.* [20] designed an "energy-efficient index tree" (EGF-tree) to save the energy utilized in collecting, querying, and aggregating data from sensors located in multiple regions in IoT. They proposed a technique to construct an EGF-tree that organizes the sensor nodes into a tree structure. The tree forwarded the queries from the sensor nodes to a base station and the results of queries from the base station to sensor nodes in an energy-efficient way. In 2014, Tang *et al.* [21] presented a similar method to construct a "clustering index tree" (ECH-tree). The IoT region was divided into grid cells that were arranged in a hierarchical manner to form a tree. Energy is saved by transmitting the data only when there is a significant change between the currently detected value and the previously sent value. In 2014, D'Oro *et al.* [22] proposed a method that exploits the fact that many "objects" move together when they are carried by a vehicle or a human being. Hence, the authors used group formation and spatial correlation for reducing the energy consumption in an IoT system. In 2013, Liang *et al.* [23] also proposed a method to save energy of the user equipment. The method allowed the user equipment to switch to sleep mode during their nonactivity period and wake up when required. The authors discussed a strategy to prolong the sleep period of the sensors for better energy efficiency. In 2013, Qiu *et al.* [24] proposed an improved GEAR network protocol, which not only reduced the paralysis rate of the network but also improved the energy utilization of the network. The authors used a multipath strategy to organize the nodes in an energy-efficient and fault-tolerant way. In addition, the authors in [25]–[29] presented energy-efficient techniques in various IoT applications. It is observed that each of the energy-saving techniques proposed in all of the aforementioned studies is applicable only in one of the IoT scenarios and that none of them is able to make the whole IoT system energy efficient.

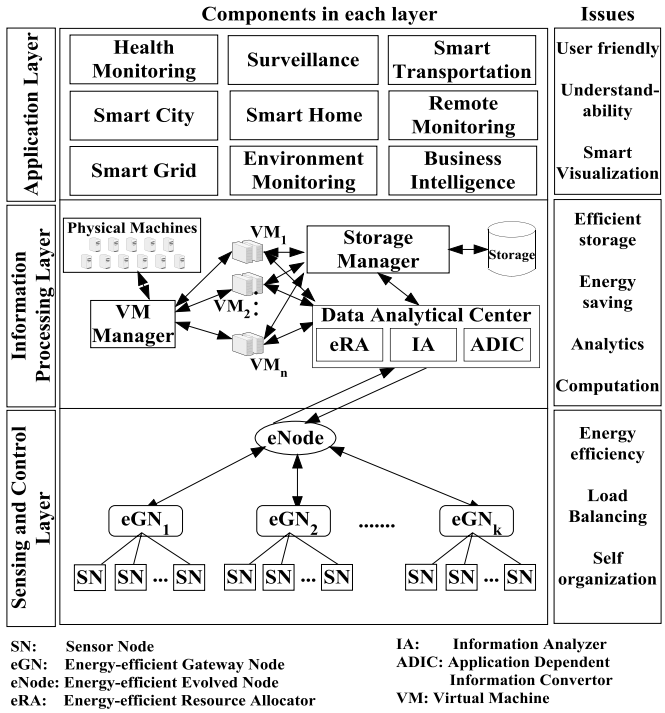

Fig. 2. Energy-efficient architecture for IoT.

Furthermore, since the WSN is the backbone of IoT systems, it is necessary to explore the work related to energy efficiency in WSNs. In 2014, Rault *et al.* [30] presented a survey of energy-saving techniques in WSNs. They discussed the tradeoff between the requirements of application and the extension of battery life of the sensor nodes. They presented a new way to classify the energy-saving techniques for WSNs. In 2014, Khan *et al.* [31] provided a brief survey of energy-efficient schemes used in WSNs for conserving the battery power of sensor nodes. They also presented different ways to provision energy from various energy sources such as solar energy. Numerous other authors have also worked in the field of energy efficiency for WSNs [32]–[43].

All of the aforementioned studies discussed various techniques to save energy of IoT sensors. However, to the best of our knowledge, none of the authors have turned their attention toward making the whole IoT architecture energy efficient. In addition, none of them discussed the mechanism to predict the sleep interval of the sensor nodes based upon their remaining battery level and previous usage history and to increase the resource utilization of allocated cloud resources when the corresponding sensors are in energy-saving mode.

III. PA

Fig. 2 shows the PA consisting of three layers, namely, sensing and control layer (SCL), information processing layer (IPL), and application layer (AL), along with the issues handled by each layer. The SCL collects data from the target

environment in an energy-efficient way and transmits it to the IPL, which, in turn, extracts useful information from it. The AL uses the information extracted by the IPL in various domains such as health monitoring, smart city, smart transportation, and others. These three layers are explained in detail below.

A. SCL

The SCL consists of the hardware elements of an IoT system. It collects raw data in large volumes and sends them for data analysis. The three main components of this layer are sensor nodes (SNs), energy-saving gateway nodes (eGNs), and an energy-efficient base station (also called the evolved node or eNode). Each of these components is explained below.

1) SNs: The SNs are responsible for data collection. They sense the target environment and send the sensor measurements to a gateway node (eGN). Based upon the frequency of data collection and transmission, SNs are classified as either trigger based or periodic [19]. The trigger-based SNs wait for a particular event (trigger) to occur and transmit data only when trigger is fired. On the other hand, periodic sensors collect and transmit data on regular intervals or on arrival of a query. Both types of SNs collect the data in their respective buffers, and their communication hardware sends the collected data to eGN.

In addition, the SNs are battery powered and have limited amount of energy, which is utilized when an SN is in active mode. An SN is said to be "active" if it is in high energy state and is actively sensing and transmitting data to eGN. The PA allows the SN to save its energy by turning off its transceivers and, hence, switching to a low energy state (also called sleep mode). An SN switches to sleep mode immediately after finishing the data transmission and remains in sleep mode until the eGN sends it a wake-up signal. The eGN can send a wake-up signal in three situations: first, when the sleep interval for the SN has expired; second, on arrival of a query; and third, when some other SN wants to communicate with it. In the first two cases, the SN senses the target environment to fill its buffer, whereas in the third case, the SN receives the incoming data in its buffer. The received data can be used by the SN as a trigger. The SN again switches to sleep mode after completion of the required action. The SN can even be switched off if it is not required to sense the target environment for longer duration of time or when the data sensed by it are useless. Thus, the sensors use their battery power efficiently by switching between active and sleep modes as and when required.

2) eGNs: The eGN is a major contributor for saving energy in the SCL. It not only provides storage media for sensed data but also acts as a controller of SNs connected to it. It calculates the sleep interval of each SN connected to it in two steps. In the first step, it predicts the length of the next sleep interval of an SN on the basis of its previous usage history. The next step uses this predicted value to calculate the actual sleep interval based on various factors, as explained below.

Let T_{n+1}^i be the predicted value of the $(n+1)$th sleep interval of the ith SN (SN^i) and t_n^i be the actual value of the nth sleep interval of SN^i. The eGN predicts T_{n+1}^i of each SN^i connected to it by using the *exponential average* [44] of the measured lengths of previous sleep intervals, as shown in (1). A factor Δt^i is then added to T_{n+1}^i to calculate the actual $(n+1)$th sleep interval by using (2). Thus

$$T_{n+1}^i = \alpha t_n^i + (1-\alpha) T_n^i;\ 0 \le \alpha \le 1 \tag{1}$$

$$t_{n+1}^i = T_{n+1}^i + \Delta t^i. \tag{2}$$

In (1), T_n^i contains the past history, whereas t_n^i is the most recent information. The initial value, i.e., T_0, is application dependent and is constant for each application. α is a controlling parameter for the relative weight of past history and present scenario in the prediction, and it determines how quickly the past is "aged." The value of α varies between 0 and 1, both inclusive. If $\alpha = 0$, then the most recent scenario has no effect, and the value of the next sleep interval depends upon the past history. If $\alpha = 1$, then the next sleep interval is equal to the previous interval. The value of α is determined by the type of sensor used. In periodic sensors, the past and the present scenarios are usually considered equally important, and hence, the value of α is taken closer to 0.5. In a trigger-based sensor, the past experiences are important for estimating the time of occurrence of the trigger event, and hence, the value of α is taken closer to 0.

In (2), the factor Δt^i is the change in the sleep interval of SN^i. The value of Δt^i can be zero, positive, or negative depending upon various factors that vary with the type of sensor used. In periodic sensors, the factor Δt^i depends upon various factors explained below and can be calculated by using

$$\Delta t^i = \frac{\oint(k) + \xi_i}{E_i * \text{CoV}}. \tag{3}$$

(i) *Quality of information* $(\oint)$: With shorter sleep intervals, the sensor harvests more data from the target environment, and hence, the quality of information is improved, but more battery power is consumed. Similarly, longer sleep intervals reduce the amount of sensed data, leading to decreased quality of information. In addition, longer sleep intervals save battery power, but they cannot be increased to the extent to which the quality of information extracted is considerably reduced. Therefore, to maintain a tradeoff between quality of information extracted and energy consumption of IoT systems, the eNode contacts the data analytical center to know the quality of information extracted from the data obtained from the kth eGN. The quality of information is divided into L levels ranging from level 1 (very low) to level L (very high). The value of $\oint(k)$ is obtained by dividing the level of information by the total number of levels as per

$$\oint(k) = \frac{\text{current level of information}}{\text{total number of levels}}. \tag{4}$$

For example, if there are ten levels of information, then the fifth level will have a value of $\oint(k)$ equal to 0.5. The calculated value is passed by the data analytical center of the IPL to the eNode, which is then conveyed to the eGN. The eGN, in turn, decides on the sleep interval depending upon the information provided by the eNode.

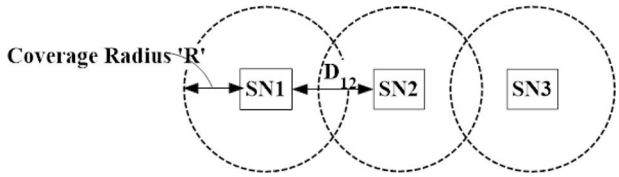

Fig. 3. SNs and their coverage area.

(ii) *Conflict factor* (ξ_i): Sometimes, the coverage area of two or more sensors may overlap. Fig. 3 represents such a case, where the dotted line represents the coverage area of each node. Here, the area covered by SN2 is in conflict with the coverage area of SN1 and SN3. The conflict factor (ξ_i) calculates the amount of the overlapped region of the ith node by using

$$\xi_i = \sum (2R - D_{ik}) \tag{5}$$

where R is the coverage radius of SNs, and D_ik is the distance between the ith and the kth node.

(iii) *Battery level* (E_i): It corresponds to the remaining battery level of SN^i. The battery level can vary from very low (less than 20%) to very high (greater than 80%). When the battery level is high, the sleep interval can be shortened. On the other hand, with the decrease in battery level, the sleep interval can be gradually increased for energy efficiency.

(iv) *Coefficient of Variation* (CoV): The eGN monitors the amount of deviation between the current and the previously sensed values. If the values do not show considerable variation, then the sleep interval can be prolonged for better energy utilization. On the other hand, if the value largely deviates, then the sleep interval is decreased to monitor the target environment efficiently. The CoV is calculated by using

$$\text{CoV} = \frac{\sqrt{\Sigma (x_i - \mu)^2}}{\mu} \tag{6}$$

where x_i's are the sensed values, and $\mu = \Sigma_{i=1}^{n} x_i / n$ is the mean of the sensed values.

In trigger-based sensors, the value of Δt^i depends upon three factors explained below and is calculated by using

$$\Delta t^i = \frac{\Delta \oint + \xi_i}{E_i}. \tag{7}$$

(i) *Conflict factor* (ξ_i)*:* It is similar to the case discussed on periodic sensors with the difference that the conflict area is decided by taking the place of occurrence of the trigger event into consideration. The SN covering the critical area in terms of occurrence of the trigger event is given more priority, and its sleep interval is considerably reduced, and vice versa. If p_i gives the probability of nonoccurrence of the trigger event in the coverage area of SN_i, then ξ_i can be calculated as per

$$\xi_i = \sum \left((2R - D_{ik})^* p_i \right). \tag{8}$$

(ii) *Battery level* (E_i): The battery level forms a less important factor in the decision of sleep interval of the trigger-based nodes because it is more important to sense the trigger event than to save energy. However, the battery level cannot be ignored altogether. The sleep interval can be increased if the probability of occurrence of the trigger event is negligible and the battery level is low.

(iii) *Effect of the trigger event on the system* $(\Delta \oint)$: Let a trigger event occur when the system is in state "s." The data analytical center measures the change in the quality of information $(\Delta \oint)$ after the occurrence of each trigger event. If there is no significant improvement in the quality of information, then the effect of the trigger event is said to be negligible. The data analytical center saves the state "s" and communicates it to the eGN via the eNode. The eGN stores all such states in its memory because if the next time trigger event occurs under such state, then its effect is more likely to be negligible. Hence, on occurrence of the next trigger event, the eGN compares the current system state with all the states stored in its memory, and if a match is found, the trigger event is ignored. This mechanism will help prolong the battery life of the trigger-based node.

After calculating the actual sleep interval of every SN connected to it, the eGN sends a wake-up signal to the SNs on expiry of their sleep intervals. Moreover, any communication between SNs is carried out through eGNs. The communication between two SNs is termed as critical when an immediate action is required by the receiving SN on message reception. However, if the communicated message contains information that can be used by the receiving SN in the future, then the communication is said to be noncritical. The sender node can indicate the type of communication by embedding a bit in the message header.

When SNs want to communicate with each other, the eGN checks whether the receiving SN is active. If the receiving SN is in sleep mode and the communication is critical, the eGN immediately sends a wake-up signal to it to ensure that there is no packet loss when the communication is initiated. However, if the communication is not critical enough, then the eGN saves the message of the sending SN in its buffer and forwards it to the receiving SN when it wakes up. In addition, it is the eGN that sends a wake-up signal to the SN (if it is in sleep mode) when a query arrives. Thus, the eGN helps the SNs to effectively utilize their energy by switching to sleep mode whenever they are idle according to Algorithm 1.

3) Energy-Efficient eNode: The SCL contains a base station or an eNode that controls all the eGNs. It fetches any required information [such as quality of information $(\oint)$] from the cloud data analytical center and passes it to eGNs. It also transmits all the data harvested by SNs to the IPL and, hence, sets up a communication path between sensors and cloud resources. In addition, the eNode allocates SNs to each eGN depending upon the battery level of the eGN and the distance of the SN from the eGN.

Let $d(i,k)$ be the distance between SN SN_i and gateway node eGN_k. Let E_k be the battery level of eGN_k, then the allocation factor $(A(i,k))$ of eGN_k with respect to SN_i can be calculated by using (9). The eNode allocates SN_i to that eGN that has maximum allocation factor $A(i,k)$, i.e.,

$$A(i,k) = \frac{E_k}{d(i,k)}. \tag{9}$$

Algorithm 1 Working of eGN

Step 1: For each SN (SN_i) in active mode do in parallel
- *Step 1.1*: Collect data from SN_i and store in its buffer
- *Step 1.2*: Send query to eNode to fetch value of α and $\oint$
- *Step 1.3*: Predict the next sleep interval T^i_{n+1} of SN_i as

$$T^i_{n+1} = \alpha t^i_n + (1-\alpha) T^i_n$$

- *Step 1.4*: Calculate the actual next sleep interval

$$t^i_{n+1} = T^i_{n+1} + \Delta t^i$$

- *Step 1.5*: If SN_i is periodic sensor
  - *Step 1.5.1*: $\xi_i = \sum(2R - D_{ik})$
  - *Step 1.5.2*: $\text{CoV} = \frac{\sqrt{\sum(\boldsymbol{x_i}-\boldsymbol{\mu})^2}}{\boldsymbol{\mu}}$; where x_i's are the sensed values and $\boldsymbol{\mu} = \frac{\sum_{i=1}^{n} \boldsymbol{x_i}}{\boldsymbol{n}}$
  - *Step 1.5.3*: $\Delta t^i = \frac{\oint(k)+\xi_i}{E_i * \text{CoV}}$
- *Step 1.6*: Else
  - *Step 1.6.1*: $\xi_i = \sum((2R - D_{ik})^* p_i)$
  - *Step 1.6.2*: $\Delta t^i = \frac{\boldsymbol{\Delta} \oint + \boldsymbol{\xi_i}}{\boldsymbol{E_i}}$

Step 2: For each sensor, SN_i, in sleep mode do
- *Step 2.1*: Send a wake-up signal on expiry of sleep interval

Step 3: For every communication request reaching eGN for SN_i
- *Step 3.1*: If communication is critical
  - *Step 3.1.1*: If sensor status = "sleep" send a wake-up signal to SN_i
  - *Step 3.1.2*: Initiate the communication
- *Step 3.2*: Else
  - *Step 3.2.1*: Buffer the information and send it when SN_i wakes up

Step 4: For every query reaching eGN for SN_i
- *Step 4.1*: If sensor status = "sleep"
  - *Step 4.1.1*: Send a wake-up signal to SN_i

In addition, the eNode keeps track of the battery level of each eGN. It classifies the amount of remaining battery energy into three levels: high ($> 70\%$), medium ($> 40\%$ and $\leq 70\%$), and low ($\leq 40\%$). Whenever the battery level of some eGN_i shifts to a lower level, the eNode again calculates the allocation factor for each gateway. It shifts "x" number of SNs connected to eGN_i to eGN_k, where eGN_k is the gateway with the maximum allocation factor, and "x" is determined by using

$$x = \frac{\text{Difference in number of SNs connected to eGN}_i \text{ and eGN}_k}{A(k)}. \tag{10}$$

This way, the eNode balances the load of each eGN for the effective utilization of energy of all the gateway nodes. Moreover, the eNode also calculates the maximum amount of data ($D_{\max}$), which can be generated during the next Δt time period depending upon the number of active sensors during Δt. The number of active sensors can be determined from their sleep intervals. If n is the number of active sensors during Δt and a is the average data generated by one SN in one time period, then $D_{\max} = a^* n$. The eNode sends $D_{\max}$ to the data analytical center so that it can allocate cloud resources accordingly.

B. IPL

The data collected by the SNs are in raw form and in large volumes. These data need to be stored, processed, and analyzed to extract interpretable information from it. This task is accomplished by the IPL, which uses storage and data analytical tools provided by the cloud computing platform. This layer consists of a data analytics center, storage media, and different physical and virtual machines. The data analytical center further consists of an energy-efficient resource allocator (eRA), an information analyzer (IA), and an application-dependent information convertor (ADIC). Each of the components of the IPL is explained below.

1) eRA: The eRA is the energy-saving component of the IPL. It allocates the hardware resources for data processing according to the requirements of the SCL. The eNode communicates the value of $D_{\max}$ to eRA, which is used to adjust the energy of allocated resources. Thus, the resources of the SCL and the IPL are "tuned" together for better performance and energy saving. This mechanism allows for the energy-efficient use of the cloud resources.

2) IA: The IA calculates the level of information extracted from data collected by the SCL. It classifies the level of information into L levels ranging from very low (level 1) to very high (level L). The factor $\oint$ (quality of information) is calculated by the IA and is conveyed to the eNode whenever the level shifts from a higher to a lower level or on reception of a query from the eNode.

3) ADIC: The information extracted by the IPL can be used by any application in the AL. The same information can be used by various applications but in different forms. ADIC converts the information into the required form for different applications. This mechanism provides application developers a platform (PaaS) for easy application development.

4) Physical/Virtual Machines and Storage Media: The physical machines at the IPL are clustered to form virtual machines by the virtual machine manager. The virtual machines, hence formed, are then used for data processing. They convert the data into an interpretable form and store them in the storage media, which is controlled by a storage manager.

C. AL

The AL provides services to the end users. It provides an interface to users for applications such as health monitoring, smart city, smart transportation, environment monitoring, surveillance, business intelligence, smart grid, and remote monitoring. The information derived from raw data collected by the sensors can be used by any application. In addition, this layer also provides the visualization tools to show the processed data.

TABLE I
INITIAL SYSTEM STATE

Parameter	Value
$\mathscr{f}$	5
CoV	0.6
ξ	0
E	80%
t_n	12 min
T_n	6 min
α	0.5
Active time interval	5 min
$\boldsymbol{E_s}$	0.002% of the remaining battery power
$\boldsymbol{E_a}$ (With sleep intervals)	0.4% of the remaining battery power
$\boldsymbol{E_a}$ (Without sleep intervals)	0.3% of the remaining battery power

Thus, the PA effectively saves energy of the hardware resources of the SCL and the IPL. The saved battery level of sensors helps prolong the lifetime of the IoT system. In a nutshell, energy efficiency is achieved by the eGN and the eNode at the SCL and by the eRA at the IPL of IoT architecture.

IV. THEORETICAL ANALYSIS

Before presenting the theoretical analysis of the system, we first explain the procedure in calculating the energy level of a node. It is observed that the energy consumption of a particular node in the system is inversely proportional to its sleep interval, which, in turn, depends upon various factors such as remaining battery level, conflict factor, quality of information, effect of trigger event, and CoV. The more the sleep interval, the lesser the energy consumption, and vice versa. Moreover, every node consumes a specified amount of energy in active mode (E_a) and in sleep mode (E_s). The amount of energy consumed in active and sleep modes is node dependent. Hence, by calculating the sleep interval using various factors, the energy level of a node can be determined by using

$$\text{Energy consumed} = T_s * E_s + (T - T_s) * E_a. \tag{11}$$

Here, T_s is the sum of all the sleep intervals, and T is the total time elapsed. Hence, $(T - T_s)$ denotes the total time for which the node is in active mode.

For the theoretical analysis of the PA, a case study is taken where initial conditions for the system are depicted in Table I. Here, active time interval is the time for which the SN is in active mode. Let this time be constant and equal to 5 min. After 5 min, the sensor switches to sleep mode for a time interval that is calculated by using (1)–(6). The value of E_a (with sleep intervals) is equal to 0.4%, which is taken in such a way that the extra energy required for state transition from sleep mode to active mode is also included. However, if the sensor continuously remains in active mode (i.e., there is no sleep interval), then $E_a = 0.3\%$ of the remaining battery power, since there are no state transitions.

The effect of the PA on the battery level of a periodic sensor is calculated and shown in Table II. Note that superscript "i" has been omitted in the notations in Table II because only one node is considered here. It is observed from the results that in 120 min, the sensor saved $((61.85 - 44)/100 =)17.85\%$ of the battery power by using the PA.

TABLE II
THEORETICAL RESULTS

Time Elapsed (min)	T_{n+1} *(min)*	t_{n+1} *(min)*	*E (with sleep interval)*	*E (without sleep interval)*
0	--	--	80	80
5	9	9.1	78	78.5
14.1	--	--	77.9818	75.77
19.1	9.05	9.159675	75.9818	74.27
28.25968	--	--	75.96348	71.5221
33.25968	9.104838	9.217506	73.96348	70.0221
42.47718	--	--	73.94505	67.25685
47.47718	9.161172	9.277001	71.94505	65.75685
56.75418	--	--	71.92649	62.97375
61.75418	9.219086	9.338259	69.92649	61.47375
71.09244	--	--	69.90782	58.67227
76.09244	9.278673	9.401388	67.90782	57.17227
85.49383	--	--	67.88901	54.35185
90.49383	9.34003	9.466506	65.88901	52.85185
99.96034	--	--	65.87008	50.0119
104.9603	9.403268	9.533741	63.87008	48.5119
114.4941	--	--	63.85101	45.65178
119.4941	9.468505	9.603237	61.85101	44.15178
120	--	--	61.85	44

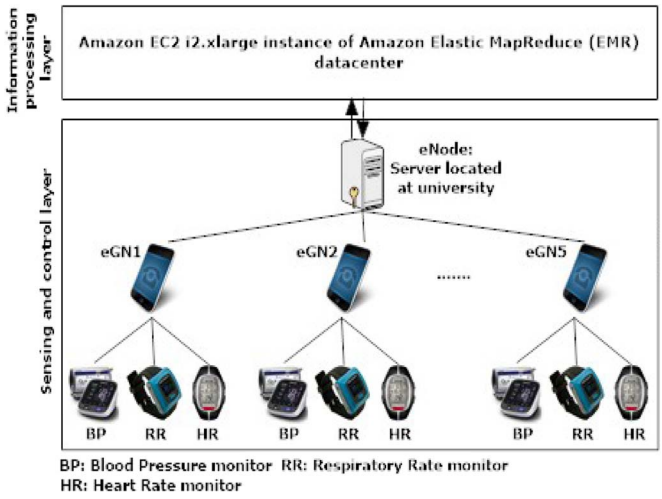

Fig. 4. Experimental setup for the PA.

V. EXPERIMENTAL SETUP AND PERFORMANCE ANALYSIS

This section discusses the experimental analysis of the PA. The experimental setup is divided into two parts: 1) initialization at the SCL and 2) data transmission to cloud environment.

A. Initialization at SCL

Fig. 4 shows the experimental setup used to evaluate the PA in a university campus. Here, five voluntary individuals from the university campus are connected to three sensors each, for observing their blood pressure (BP), heart rate (HR), and respiratory rate (RR). The sensors used to monitor the BP, HR, and RR of these individuals are 1) Omron 10 upper arm BP monitor model BP785 [45], 2) Polar RS 300× HR monitor [46], and 3) CMS 50F OLED wrist RR monitor [47]. Tables III and IV list the details of the individuals and sensors, respectively.

TABLE III
DETAILS OF FIVE VOLUNTARY INDIVIDUALS FROM UNIVERSITY CAMPUS

S.No.	*Age (yrs)*	*Height (cm)*	*Weight (kg)*	*Medical history*
1	19	168	68	None
2	26	176	75	Hypertension
3	34	165	72	Asthma
4	47	172	89	None
5	62	170	83	Heart Patient

TABLE IV
AMOUNT OF DATA GENERATED BY EACH SENSOR

Sensor	*Data generated*
Omron 10 upper arm BP monitor	52KB per10 sec
Polar RS 300x HR monitor	108 KB per 10sec
CMS 50F OLED wrist RR monitor	143 KB per trigger

The first two sensors are treated as periodic sensors, and the third sensor is treated as a trigger-based sensor that triggers when the HR increases above a certain limit. It is assumed that the conflict factor of each sensor is zero. These sensors are connected through Bluetooth to the Android-based mobile phones of each individual. The mobile phone acts as the eGN of the PA. A JavaScript-based application is installed on each mobile phone to run the energy-efficient algorithm for the eGN. The server located in the university campus acts as the eNode of the PA. The eNode sends the sensor data to the Amazon Elastic MapReduce (Amazon EMR) datacenter [48] hosting Hadoop v1.0.3 clusters.

In a 120-min experiment, the performance of the sensors, i.e., eGN and eNode, is measured both with and without sleep intervals and is shown in Fig. 5(a)–(e). The variation of sleep interval with the variation of α is shown in Fig. 5(f). Fig. 5(g) and (h) shows the variation of factor Δt with the decreasing battery level for different information levels and CoV values, respectively. The dependence of the actual sleep interval on the predicted sleep interval and Δt with the variation of battery level is shown in Fig. 5(i).

B. Data Transmission to Cloud Environment

Since data generated by the sensors are less, the bootstrapping technique [49] is used. The data from five users are bootstrapped to 5000 users using correlation coefficient as the variant. A 95% confidence interval for the correlation coefficient between age and BP is [0.3319 0.9427], that between age and HR is [0.4526 0.9348], and that between HR and RR is [0.5692 0.9436]. The bootstrapped data are sent to cloud. Amazon EC2 storage optimized i2.xlarge instance [50] is used to test the performance of the PA. The response time and resource utilization of cloud resources is measured and is shown in Fig. 5(j) and (k), respectively. To check the scalability of the PA, the experiment is repeated with varying numbers of nodes. Fig. 5(l) shows the results.

C. Results and Discussion

Fig. 5(a)–(d) shows the performance of the three sensors and the eGN with respect to the battery level. It is observed that there is an energy-efficient consumption of battery power using sleep intervals because energy is saved by switching the sensors and the eGN to low energy mode based on various factors, as described in Section III. In the case of an RR sensor, the sudden decrease in battery level shown in Fig. 5(b) corresponds to a trigger event. Fig. 5(e) shows the energy efficiency of the eNode. It can be observed that, in the case of the system with no sleep intervals, there is a sudden consumption of energy by the eNode in the beginning. This is because energy is consumed to configure all the active nodes at the start. Thereafter, all the nodes continuously keep on sending data without switching to sleep mode, and hence, the eNode consumes almost the same amount of energy in the rest of the time period. On the other hand, when the nodes switch to sleep mode, the eNode receives comparatively smaller amounts of data due to which it consumes lesser energy. When a sensor switches its state from sleep mode to active mode, it consumes an extra amount of energy. However, the energy saved by the PA is more than the extra energy consumed by the system due to state transition. The results in Fig. 5(a)–(e) support the above inference.

Fig. 5(f) shows the effect of α on predicted sleep interval. The trends in the graph show that lower values of α favor the most recent sleep interval and, hence, show a flatter curve. On the other hand, larger values of α lead to larger variation in sleep interval due to high weightage given to past history.

Fig. 5(g) illustrates the variation of Δt with the decreasing battery level and different levels of $\oint$, keeping $\text{CoV} = 0.5$ at a constant value. It is observed that when the quality of information is very low, the sleep interval is almost zero so that more data can be harvested from the target environment, and hence, the quality of information can be improved. The sleep interval increases with the increase in quality of information and with the decrease in battery level because if the quality of information is higher, then the sleep interval can be effectively increased without much variation in the quality of information.

Fig. 5(h) represents the trend followed by Δt with different values of CoV, keeping $\oint = 5$ at a constant value. The results show that Δt rapidly increases with the decreasing battery level. This increase in Δt with the decreasing battery power increases the sleep interval of an SN, which is desirable for energy saving. Moreover, the increase is higher for lower values of CoV because if there is a large variation in the consecutively sensed values, then the sleep interval cannot be rapidly increased, and vice versa. Fig. 5(i) shows the variation of Δt, predicted sleep interval, and actual sleep interval with the decreasing battery level of the sensors. The results show that the actual sleep interval initially follows the trend of predicted sleep interval because there is a surplus amount of battery power available. However, as the battery level drops below 50%, the actual sleep interval started following the trend of Δt because with the decreased battery level, the sleep interval is increased by lowering the quality of information.

The performance of the PA in cloud environment is shown in Fig. 5(j) and (k). It is observed that there is a considerable improvement in terms of response time and resource utilization of cloud resources. This is so because cloud resources are switched to sleep mode or are reprovisioned based upon the

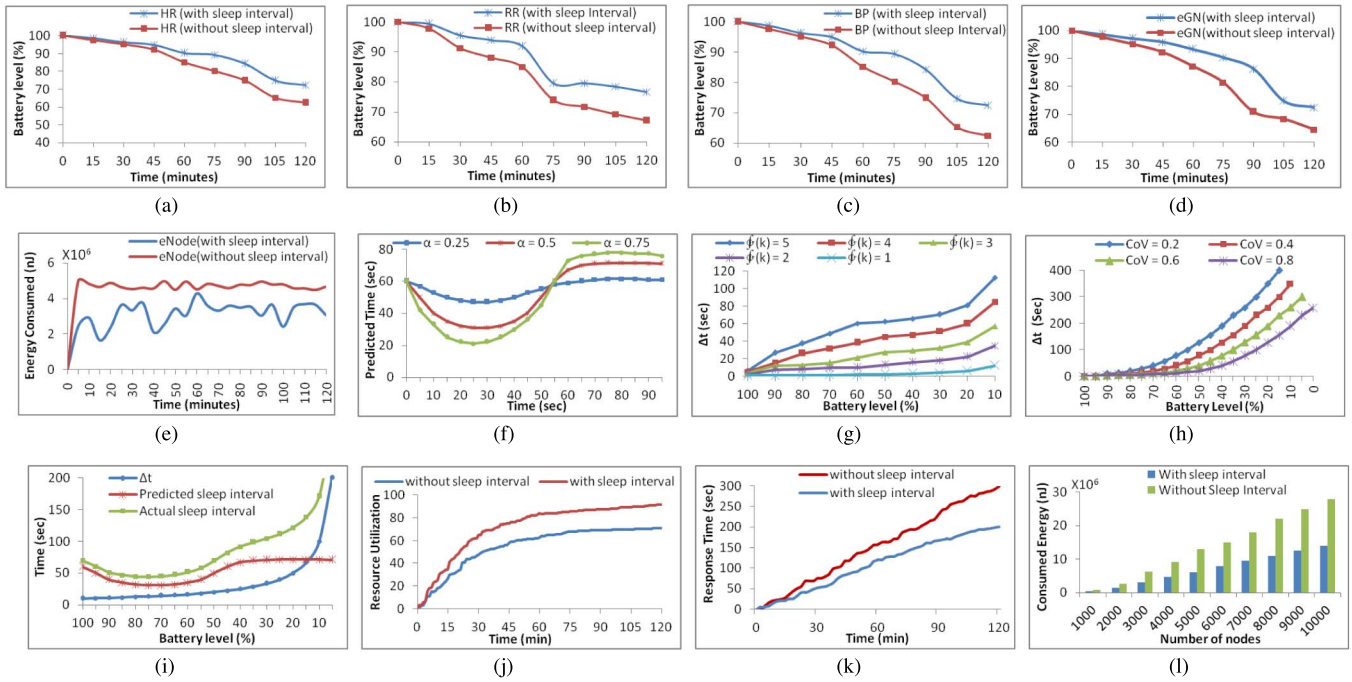

Fig. 5. (a)–(l) Experimental results: (a)–(c) variation of the battery levels of HR, RR, and BP monitors, respectively, when they sense data with and without sleep intervals; (d) variation of the battery levels of the eGN, both with and without sleep intervals; (e) energy consumed by the eNode with and without sleep intervals; (f) variation of predicted sleep intervals with the variation of α; (g) and (h) variation of Δt with quality of information and CoV, respectively; (i) dependence of actual sleep interval on predicted sleep interval and Δt with the variation of battery levels; (j) and (k) variation of response time and resource utilization, respectively, of cloud resources, both with and without sleep intervals; (l) variation of energy consumed with respect to the number of nodes.

value of $D_{\max}$. Thus, the cloud resources are utilized in an energy-efficient way. Fig. 5(l) shows the scalability of the PA with varying numbers of nodes. The results show that, with the increasing number of nodes, a lower amount of energy is consumed by the PA.

VI. Comparative Analysis

This section is aimed toward comparing the performance of the PA in three cases: 1) functionality comparison with related techniques; 2) experimental comparison with related techniques; and 3) performance comparison of the PA under various scenarios and system settings. The first two cases compare the PA with some other energy efficiency methods discussed in Section II. The methods used for comparison are SoT [19], EGF-tree [20], ECH-tree [21], and object group localization (OGL) [22]. The third case highlights the suitability of the PA in various scenarios.

A. Functionality Comparison

Table V shows the functionality comparison of the PA with other techniques. It can be observed from the Table that 1) the energy efficiency of the cloud resources is considered only in the PA and that 2) the energy efficiency of SNs is well thought out in SoT, ECF, and the PA, whereas EGF and OGL do not work on this issue.

1) The PA allows nodes to move in the network since the calculation of the sleep interval and all the related factors is dynamic. OGL is specifically designed for mobile IoT networks. Contradictorily, other methods do not allow nodes to move in an IoT network.
2) The remaining battery level of the SNs is a key element while calculating the sleep interval in the PA. None of the other techniques took battery level into consideration.
3) All the methods except SoT scale well with varying numbers of nodes in an IoT network.
4) All the methods, other than the PA, are applicable only in some of the IoT scenarios. On the other hand, since an architecture is proposed in this paper (rather than a method), it can be applied in a vast number of IoT networks.

From the above, it can be concluded that the PA shows a functionality edge over other techniques.

B. Experimental Comparison

Fig. 6(a) shows the performance comparison of the PA with SoT, EGF, ECH, and OGL. The PA showed performance improvement over other methods by consuming lesser amount of energy.

C. Scenario Comparison

It can be observed that the system settings in various real-life applications vary mainly due to the number of nodes in the system, traffic rate of the network, or density of nodes. For example, industrial IoT networks are dense and have a high traffic rate. There are large numbers of nodes in such networks. On the other hand, smart home consists of low traffic rate and less number and density of nodes. Similar is the case with other real-life applications. Hence, to check the applicability of the PA in various scenarios, the system is tested by varying traffic rate, node density, and number of nodes. The traffic rate is

TABLE V
FUNCTIONALITY ANALYSIS

S.No.	Parameters	SoT	EGF	ECF	OGL	Proposed Architecture
1.	Energy efficiency of cloud resources	No	No	No	No	Yes
2.	Energy efficiency of SNs	Yes	No	Yes	No	Yes
3.	Node Mobility	No	No	No	Yes	Yes
4.	Battery level consideration	No	No	No	No	Yes
5.	Scalability	No	Yes	Yes	Yes	Yes
6.	Applicability	Dense IoT network with low traffic rate	Data collection and aggregation for query handling in IoT	Handle time correlated region queries	Mobile IoT networks	Many IoT networks
7.	Methodology used	Sleep scheduling	EGF- Tree	ECH-Tree	Object Group Mobility	Sleep scheduling based architecture
8.	Self learning mechanism	Yes	Yes	No	Yes	Yes

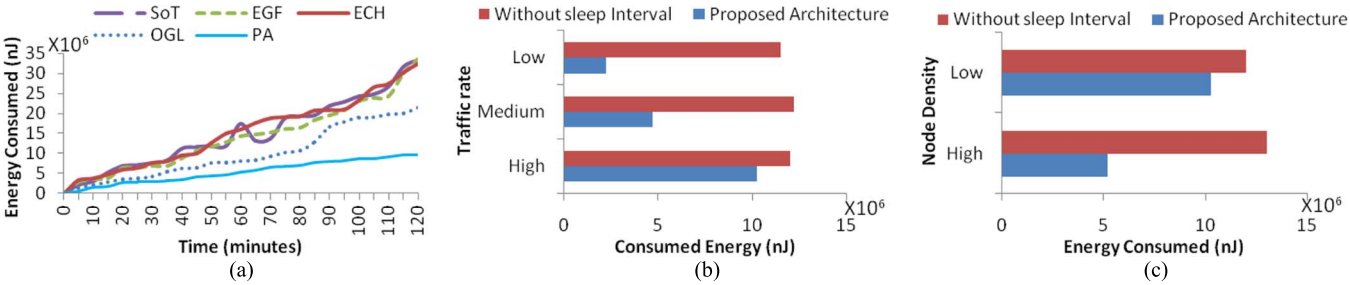

Fig. 6. (a)–(c) Comparative results: (a) performance comparison of the PA with SoT, EGF-tree, ECH-tree, and OGL; (b) and (c) amount of energy consumed by the system with 4500 nodes with various levels of traffic rate and node density, respectively.

varied by varying the number of times an SN sends data and messages to the eGN and other SNs, respectively. The more the data communicated, the more the traffic rate, and vice versa.

Therefore, to test the suitability of the PA to various system settings and scenarios, an experiment is performed by varying the number of nodes, traffic rate of the network, and density of nodes in the system. The performance of the system with an increased number of nodes has already been discussed in the previous section. The results depicting the amount of energy consumed by a 4500-node system with different traffic rates and node densities are shown in Fig. 6(b) and (c), respectively.

Fig. 6(b) shows the amount of energy consumed by the system with high, medium, and low network traffic. The results show that the PA achieves the best performance when the traffic rate is low. This is because low traffic allows larger sleep intervals, leading to better performance. On the other hand, a higher traffic rate decreases the sleep interval of nodes, leading to relatively higher energy consumption. It can also be noted that even at a high traffic rate, the energy consumption is low for the PA than without the use of sleep interval.

The energy consumed by the system also varies with the density of nodes. Fig. 6(c) shows the result. The results depict that the higher the node density, the lower the energy consumed. This is because higher node density leads to a higher conflict factor among the nodes, and hence, more nodes are switched to sleep mode, which, in turn, results in lower energy consumption. From all the above results, it can be concluded that the PA for IoT is energy efficient and is applicable in all types of IoT networks.

VII. CONCLUSION

In this paper, architecture for IoT has been proposed, which ensures an energy-efficient utilization of the resources. The architecture is tested by using medical data on Amazon EC2 i2.xlarge instance. The results show that the energy is effectively and efficiently saved by switching the hardware resources of the SCL and the IPL to sleep mode. The key feature of the proposed model is the exchange of energy-related information between the two layers. The sensors switch to sleep mode based upon their available battery power and other factors, such as quality of extracted information, conflict factor, and CoV. This mechanism enables cloud environment to predict the maximum amount of data that can be received during the next time interval, and hence, resources can be provisioned accordingly. Hence, the PA effectively increased resource utilization of hardware resources of both the SCL and the IPL. In a nutshell, the PA is energy efficient. Moreover, due to the flexile nature of the PA, it can be applied in a large number ofIoT networks.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[2] R. Caceres and A. Friday, "Ubicomp systems at 20: Progress, opportunities, and challenges," *IEEE Pervasive Comput.*, vol. 11, no. 1, pp. 14–21, Jan./Mar. 2012.

[3] Z. Bi, L. D. Xu, and C. Wang, "Internet of things for enterprise systems of modern manufacturing," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1537–1546, May 2014.

[4] S. Fang *et al.*, "An integrated system for regional environmental monitoring and management based on Internet of things," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1596–1605, May 2014.

[5] D. Zhang *et al.*, "Real-time locating systems using active RFID for Internet of things," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2346625, to be published.

[6] F. Tao, Y. Zuo, L. D. Xu, L. Lv, and L. Zhang, "Internet of things and BOM-based life cycle assessment of energy-saving and emission-reduction of products," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1252–1261, May 2014.

[7] F. Tao, Y. Cheng, L. D. Xu, L. Zhang, and B. H. Li, "CCIoT-CMfg: Cloud computing and Internet of things-based cloud manufacturing service system," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1435–1442, May 2014.

[8] F. Tao, Y. Zuo, L. D. Xu, and L. Zhang, "IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1547–1557, May 2014.

[9] L. Liu, W. Han, T. Zhou, and X. Zhang, "SCout: Prying into supply chains via a public query interface," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2337519, to be published.

[10] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[11] F. Paganelli, S. Turchi, and D. Giuli, "A web of things framework for RESTful applications and its experimentation in a smart city," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2354835, to be published.

[12] E. Patti *et al.*, "Event-driven user-centric middleware for energy-efficient buildings and public spaces," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2302750, to be published.

[13] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the Internet of things," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2298837, to be published.

[14] J. Chen, B. Wang, W. Liu, L. T. Yang, and X. Deng, "Rotating directional sensors to mend barrier gaps in a line-based deployed directional sensor network," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2327793, to be published.

[15] A. H. Celdran, F. J. Garcia Clemente, M. G. Perez, and G. M. Perez, "SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2013.2297707, to be published.

[16] Y. Liang *et al.*, "An integrated approach of sensing tobacco-oriented activities in online participatory media," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2304706, to be published.

[17] Y. Hu, M. Dong, K. Ota, A. Liu, and M. Guo, "Mobile target detection in wireless sensor networks with adjustable sensing frequency," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2308391, to be published.

[18] J. Pan *et al.*, "A Internet of things framework for smart energy in buildings: Designs, prototype, and experiments," *IEEE Internet Things J.*, DOI: 10.1109/JIOT.2015.2413397, to be published.

[19] O. U. Akgul and B. Canberk, "Self-Organized Things (SoT): An energy efficient next generation network management," *Comput. Commun.*, DOI: 10.1016/j.comcom.2014.07.004, to be published.

[20] Z. Zhou, J. Tang, L. Zhang, K. Ning, and Q. Wang, "EGF-tree: An energy-efficient index tree for facilitating multi-region query aggregation in the Internet of things," *Pers. Ubiquitous Comput.*, vol. 18, no. 4, pp. 951–966, Apr. 2014.

[21] J. Tang, Z. Zhou, J. Niu, and Q. Wang, "An energy efficient hierarchical clustering index tree for facilitating time-correlated region queries in the Internet of things," *J. Netw. Comput. Appl.*, vol. 40, pp. 1–11, Apr. 2014.

[22] S. D'Oro, L. Galluccio, G. Morabito, and S. Palazzo, "Exploiting object group localization in the Internet of things: A performance analysis," *IEEE Trans. Veh. Technol.*, vol. 64, no. 8, pp. 3645–3656, Aug. 2015.

[23] J. Liang, J. Chen, H. Cheng, and Y. Tseng, "An energy-efficient sleep scheduling with QoS consideration in 3GPP LTE-advanced networks for Internet of things," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 3, no. 1, pp. 13–22, Mar. 2013.

[24] T. Qui, W. Sun, Y. Bai, and Y. Zhou, "An efficient multi-path self-organizing strategy in Internet of things," *Wireless Pers. Commun.*, vol. 73, no. 4, pp. 1613–1629, Dec. 2013.

[25] M. A. Hoque, M. Siekkinen, and J. K. Nurminen, "Energy efficient multimedia streaming to mobile devices—A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 579–597, 1st Quart. 2014.

[26] V. Sai and M. H. Mickle, "Exploring energy efficient architectures in passive wireless nodes for IoT applications," *IEEE Circuits Syst. Mag.*, vol. 14, no. 2, pp. 48–54, 2nd Quart. 2014.

[27] F. Lin, Q. Liu, X. Zhou, Y. Chen, and D. Huang, "Cooperative differential game for model energy–bandwidth efficiency tradeoff in the Internet of things," *China Commun.*, vol. 11, no. 1, pp. 92–102, Jan. 2014.

[28] C. Hou and Q. Zhao, "Bayesian prediction-based energy-saving algorithm for embedded intelligent terminal," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, DOI: 10.1109/TVLSI.2014.2385791, to be published.

[29] C. H. Liu, J. Fan, J. W. Branch, and K. K. Leung, "Toward QoI and energy-efficiency in Internet-of-things sensory environments," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 4, pp. 473–487, Dec. 2014.

[30] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Comput. Netw.*, vol. 67, pp. 104–122, Jul. 2014.

[31] J. A. Khan, H. K. Qureshi, and A. Iqbal, "Energy management in wireless sensor networks: A survey," *Comput. Elect. Eng.*, vol. 41, pp. 159–176, Jan. 2015.

[32] K. Lin, J. J. P. C. Rodrigues, H. Ge, N. Xiong, and X. Liang, "Energy efficiency QoS assurance routing in wireless multimedia sensor networks," *IEEE Syst. J.*, vol. 5, no. 4, pp. 495–505, Dec. 2011.

[33] H. K. Deva Sarma, R. Mall, and A. Kar, "E2R2: Energy-efficient and reliable routing for mobile wireless sensor networks," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2015.2410592, to be published.

[34] K. Illanko, M. Naeem, A. Anpalagan, and D. Androutsos, "Energy-efficient frequency and power allocation for cognitive radios in television systems," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2015.2393834, to be published.

[35] M. M. Hasan, F. Farahmand, J. P. Jue, and J. J. P. C. Rodrigues, "A study of energy-aware traffic grooming in optical networks: Static and dynamic cases," *IEEE Syst. J.*, vol. 7, no. 1, pp. 161–173, Mar. 2013.

[36] C. Yu, D. Yao, L. T. Yang, and H. Jin, "Energy conservation in progressive decentralized single-hop wireless sensor networks for pervasive computing environment," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2339311, to be published.

[37] K. Heussen, S. Koch, A. Ulbig, and G. Andersson, "Unified system-level modeling of intermittent renewable energy sources and energy storage for power system operation," *IEEE Syst. J.*, vol. 6, no. 1, pp. 140–151, Mar. 2012.

[38] M. de Paula Marques, F. R. Durand, and T. Abrao, "WDM/OCDM energy-efficient networks based on heuristic ant colony optimization," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2345665, to be published.

[39] G. Sun *et al.*, "Power-efficient provisioning for online virtual network requests in cloud-based data centers," *IEEE Syst. J.*, vol. 9, no. 2, pp. 427–441, Jun. 2015.

[40] V. Pal, G. Singh, and R. P. Yadav, "Balanced cluster size solution to extend lifetime of wireless sensor networks," *IEEE Internet Things J.*, DOI: 10.1109/JIOT.2015.2408115, to be published.

[41] Y. Liu, C. Xu, and S. Cheung, "Diagnosing energy efficiency and performance for mobile Internetware applications: Challenges and opportunities," *IEEE Softw.*, vol. 32, no. 1, pp. 67–75, Jan./Feb. 2015.

[42] H. P. Gupta and S. V. Rao, "Demand-based coverage and connectivity-preserving routing in wireless sensor networks," *IEEE Syst. J.*, DOI: 10.1109/JSYST.2014.2333656, to be published.

[43] U. Kulau, F. Busching, and L. Wolf, "Undervolting in WSNs—Theory and practice," *IEEE Internet Things J.*, vol. 2, no. 3, pp. 190–198, 2015.

[44] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, 8th ed. New Delhi, India: Wiley, ch. 5, pp. 189–192.

[45] Omron, Last accessed on Jul. 5, 2015. [Online]. Available: http://www.omron-healthcare.com/eu/en/our-products/blood-pressure-monitoring

[46] Polar, Last accessed on Jul. 3, 2015. [Online]. Available: http://www.polar.com/en/products/improve_fitness/running_multisport/RS300X

[47] Amazon, Last accessed on Jul. 3, 2015. [Online]. Available: http://www.amazon.com/Wrist-Color-Oximeter-Acc-SnugFit/dp/B00EGL9SC0

[48] Amazon Elastic MapReduce, Last accessed on Jul. 3, 2015. [Online]. Available: http://aws.amazon.com/elasticmapreduce/

[49] X. Bao *et al.*, "Helping mobile apps bootstrap with fewer users," in *Proc. UbiComp*, 2012, pp. 1–10.

[50] Amazon, Last accessed on Jul. 5, 2015. [Online]. Available: http://aws.amazon.com/ec2/instance-types/

Authors' photographs and biographies not available at the time of publication.