This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2692522, IEEE Access

EC-CENTRIC: An Energy- and Context-Centric Perspective on IoT Systems and Protocol Design

Alessandro Biason, Chiara Pielli, Michele Rossi, *Senior Member, IEEE,*
Andrea Zanella, *Senior Member, IEEE,*, Davide Zordan, Mark Kelly, and Michele Zorzi, *Fellow, IEEE*

***Abstract*—The radio transceiver of an IoT device is often where most of the energy is consumed. For this reason, most research so far has focused on low power circuit and energy efficient physical layer designs, with the goal of reducing the average energy per information bit required for communication. While these efforts are valuable per se, their actual effectiveness can be partially neutralized by ill-designed network, processing and resource management solutions, which can become a primary factor of performance degradation, in terms of throughput, responsiveness and energy efficiency. The objective of this paper is to describe an energy-centric and context-aware optimization framework that accounts for the energy impact of the fundamental functionalities of an IoT system and that proceeds along three main technical thrusts: 1) balancing signal-dependent processing techniques (compression and feature extraction) and communication tasks; 2) jointly designing channel access and routing protocols to maximize the network lifetime; 3) providing self-adaptability to different operating conditions through the adoption of suitable learning architectures and of flexible/reconfigurable algorithms and protocols. After discussing this framework, we present some preliminary results that validate the effectiveness of our proposed line of action, and show how the use of adaptive signal processing and channel access techniques allows an IoT network to dynamically tune lifetime for signal distortion, according to the requirements dictated by the application.**

***Index Terms*—Context-Awareness, Energy-Efficiency, Internet of Things, Protocol Design**

I. INTRODUCTION

THE radio transceiver of an IoT device is often where most of the energy is consumed. For this reason, most research so far has focused on low power circuit design and energy efficient PHY, with the goal of reducing the average energy per information bit required for communication. While any advances at the RF/PHY layer are expected to translate into a more energy-efficient device, it is not at all obvious that this is by itself sufficient for the whole system to make the best use of the available energy, and a more complete view of the system, including the application (signal type and processing tasks), the lower networking layers (MAC scheduling and routing) and some basic network management functionalities (node discovery and sleep modes) can play a crucial role in identifying the main sources of energy consumption, revealing inefficiencies, and providing opportunities for large gains. Even though cross-layer design and holistic system approaches have been around for some time, we believe that energy efficiency at the system level can be achieved only including in a coherent and coordinated way many different functionalities that are traditionally considered in isolation, or at most in combination with neighboring functional blocks. Toward this, in this paper we propose a framework where *learning and adaptation capabilities* are applied to data handling and processing (which include most traditional networking functionalities), to optimize the energy efficiency of the system while effectively handling the heterogeneous QoS requirements of the applications. The framework is based on the following key technical thrusts:

- *In-node data processing.* Following the “edge computing” principle, an energy-centric framework should support the shift of the computation load toward the edge of the network, i.e., to the access node (e.g., the IoT Gateway) and/or to the end nodes. In-node signal processing/compression mechanisms can indeed reduce the amount of data to be transmitted, thus relieving channel contention, transport and interference issues. The energy cost of data processing must be accounted for, so as to find an optimal balance between processing and communication;
- *Channel access, scheduling and routing optimization.* These functionalities have a major impact on the energy efficiency of the system and, hence, need to be jointly designed to prolong the network lifetime by making the best possible use of the available energy resources, also including sleep modes (duty cycling and wakeup radios) and Energy Harvesting (EH) sources. The protocol design needs to account for the heterogeneity of the nodes capabilities, in order to favor a more balanced energy consumption, e.g., through the dynamic scheduling of processing and routing functionalities to energy-rich and computation-capable nodes (e.g., the IoT Gateway), or to nodes that can access EH sources;
- *Self-adaptability.* Different applications and services pose different and, often, conflicting requirements to the IoT system. Moreover, even when the application is fixed beforehand (e.g., urban traffic control), the actual data that is collected may show different statistics, depending on the deployment location and time. These facts imply a different usage of network resources that for efficiency purposes can be learned on the fly and exploited to optimize the network protocols. The system, hence, must be able to self-adapt to the application needs, being driven by the types of data, their statistics, the node density,

Mark Kelly is with Intel Labs Europe, Leixlip, Ireland, email: mark.y.kelly@intel.com; all the other authors are with the Department of Information Engineering of the University of Padova, PD, 35131, Italy, e-mails: {firstname.lastname}@dei.unipd.it

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2692522, IEEE Access

the PHY layer parameters and so forth. On-line learning techniques are expected to play a primary role in this respect, providing effective tools to handle the vastness and complexity of the optimization space, together with flexible/reconfigurable algorithms and protocols.

The core of the proposed framework is an energy-centric and self-adapting manager, with the task of jointly optimizing the energy consumption of the main networking functionalities (data compression, scheduling, MAC, routing, etc.) under QoS constraints and in the presence of EH capabilities. The manager relies on a learning framework for context awareness and self-adaptation, which provides the necessary input to the optimization process, making it flexible and scalable. After describing the framework, in this paper we provide a concrete example of classification and adaptation in the context of channel access resource management. In this setup, multiple sources concurrently transmit their (compressible) information flows to the same gateway node and the task is to dynamically balance compression (signal processing) and channel access (scheduling) resources based on the distortion-rate relationship of the flows and on application-dependent QoS requirements.

The rest of this paper is structured as follows. In Section II we discuss the related work. The energy centric framework is introduced in Section III. In Sections IV and V we respectively describe the system setup and a joint context extraction and channel access problem. The conclusions are finally drawn in Section VI.

II. Related work

Next, we analyze the related work for the three technical areas of Section I, pointing out for each the original aspects of our approach.

Data processing. A recent trend in IoT deployments is to move some of the processing from the network center to its edge, according to the *fog computing* paradigm [1]. This load shift is driven by the gigantic amount of data that is often generated by distributed sensing applications (e.g., environmental and traffic monitoring), which is expected to be a burden to the IoT system itself and to its connected networks. This burden is likely to translate into an excessive energy consumption for the IoT nodes and as well into congestion for the communication channels and gateways. Given this, in-node and in-network data processing are of paramount importance, as they can effectively reduce the amount of information that is to be sent to (and processed by) the higher levels of IoT systems [2].

In many IoT applications, like industrial and environmental monitoring, nodes periodically report measurements to a central entity (the sink) and their data volume can be highly reduced through predictive algorithms [3], [4], i.e., by sending data points only when they deviate from some expected pattern. The effectiveness of this approach has also been proved on real datasets [5]. When dealing with time series, *lossy compression* can be exploited to trade some accuracy in the signal's representation for improved energy efficiency. In this domain, a number of approaches like probabilistic, linear or autoregressive models, Fourier transforms and Kalman filters have been considered, although they are generally too computationally expensive and, in turn, power-hungry for constrained IoT devices [6]. The research community has thus started exploring lighter algorithms, e.g., Lightweight Temporal Compression [7].

The heterogeneous and dynamic nature of IoT systems requires adaptability, and employing a traditional compression scheme may lead to suboptimal performance. The research focus is thus moving towards data-driven approaches, where the compression technique is automatically adjusted according to the type of signal and to the application requirements. For example, in [8] compressive sensing is combined with principal component analysis to capture the spatial and temporal characteristics of real signals. A feedback control loop estimates the signal reconstruction errors on the fly and allows the system to self-adapt to changes in the signal statistics. The authors of [9] propose another adaptive scheme that switches between lossless and lossy compression in an on-demand fashion according to a compression error bound (derived from the application requirements). Another promising approach consists in applying data mining techniques to extract features from time series, seeking feature-based classifiers [10]. Signal classification into groups with similar characteristics allows the sensors to choose the data processing technique that is most appropriate for their respective class (i.e., leading to the best performance for some metric, like the distortion of the compressed signal) [11], [12].

Novelty: our approach develops along the same lines of [8], [10], [11], [12]; we seek a data processing algorithm that self-adapts to the properties of the signal that is being measured, i.e., its inherent correlation but also its generation rate. We want it to be data-driven, as rate-distortion curves are estimated at runtime, solely based on what each IoT node measures, using a small portion of the data (e.g., a few hundred samples) and lightweight classifiers. Moreover, the cost (energy and distortion) of the in-node processing algorithms shall be included in the optimization of the network protocols so as to allow the entire system to adapt, seeking a good tradeoff in terms of overall energy consumption (processing and communication) *vs* quality of the information that is sent to the application (e.g., quality of an answer or representation accuracy of a measurement).

Channel access, scheduling and routing. The energy constraints that characterize most of the IoT devices demand energy-aware protocols and, although data processing already provides some savings in the energy consumption, it is not in itself sufficient to guarantee prolonged and uninterrupted operation. In this respect, channel access and transmission scheduling play a crucial role because of their influence on the usage of the energy-hungry transceiver. The design of the MAC layer should try to minimize the energy wastage due to *collisions*, the *overhead* due to control packets, *idle listening*, and *overhearing*, i.e., when a device receives a packet intended for another destination [13].

Coordinated access schemes are well suited for applications where the traffic pattern is known in advance, e.g., industrial wireless sensor networks [14]. In 2012, the Internet Engineering Task Force (IETF) introduced the Time-Slotted Channel

Hopping (TSCH) [15] mode as an amendment to the MAC portion of the IEEE802.15.4e standard, which combines time synchronization and channel hopping and is intended for industrial automation. TDMA-based schemes can be effectively coupled with duty cycling, where nodes alternate active and sleeping phases [16], or can also be combined with Channel Sensing Multiple Access (CSMA) techniques, in a hybrid approach that offers more flexibility in the choice of the frame length and in the assignment of access slots to nodes [17], [18]. However, pure coordinated access schemes may result in poor performance when dealing with event-based signals such as alarms, which have strict latency and QoS constraints. Traditional protocols should therefore be revisited in order to account for the different traffic types, like in [19], where the proposed access mechanism proactively tunes the number of used resources to meet the application requirements.

In more dynamic environments, where coordinated scheduling is costly, random access schemes are generally preferred [20]. In this case, the network designer should pay special attention to interference and collision management, so that energy wastage is minimized. An interesting approach in this area is represented by coded random access schemes, which map the structure of the access protocol to that of an erasure-correcting code defined on a graph, making it possible to achieve much better performance than simple ALOHA [21]. Another way to improve the performance of random access is to consider a receiver with signal interference cancellation or multiple packet reception capabilities [22]. Also, in the case of random access schemes, duty cycling may lead to significantly reduced energy consumption, but has an impact on data latency and still wastes energy for idle listening [23]. Wake-up receivers (WURs) are a novel hardware approach that eliminates these shortcomings: devices are provided with an ultra low power receiver that continuously listens to the channel and wakes-up the main radio on demand [24]. WURs improve the overall network's energy efficiency, but their design has to deal with several tradeoffs concerning sensitivity, resilience to interference, coverage area, wake-up speed, and power consumption [25].

A well-designed channel access scheme is not itself sufficient to ensure efficient and reliable communication in multihop networks, where network connectivity is hampered by the energy and resource limitations of the IoT devices, which in some cases may not even have enough energy to forward packets, and by the dynamic network topology [26]. The unpredictable nature of IoT networks makes flat routing protocols (e.g., [27]) a good choice because they make it possible to maintain the network structure easily, but, on the other hand, hierarchical approaches like cluster-based routing algorithms (e.g., [28]), allow nodes to take on different roles, thereby enabling the possibility to leverage on the different capabilities of the devices and facilitating the aggregation of data while it is being routed.

Current research is focusing on *context-aware* routing algorithms [29], which should readily find the best alternative path when the selected one is no longer available. Cross-layer metrics are often adopted because they provide a more comprehensive picture than the current context. For example, in [30], the Routing Protocol for Low-Power and Lossy Networks (RPL) is extended to jointly consider the residual energy levels and the expected transmission count; a similar approach is also proposed in [31]. One of the issues that routing algorithms have to face concerns unexpected link failures, and to this aim self-learning techniques may be helpful to detect sensor faults, e.g., in industrial [32] or home automation [33] environments. For example, [34] describes a routing strategy that automatically adapts to the changing network conditions and shows the effectiveness of this approach.

Novelty: we advocate channel access and routing layers that "learn" what is the preferred (time-varying) configuration based on application requirements, network topology, data type, characteristics, etc. So, the optimization does not only relate to the protocol parameters but also depends on the protocol (or combination thereof) to use. This holistic approach entails self-learning and self-adaptivity and, due to its importance, is further discussed next.

Self-adaptability. From previous sections it already emerges that IoT systems, because of their heterogeneous and dynamic nature, demand a strong adaptability to the context. Devices are expected to self-manage without external intervention, and flexibility is a major property which is deemed necessary for an efficient and resilient system. Ideally, IoT devices should autonomously learn about the physical environment in which they are deployed, learn to manage themselves and find place in the overall system, thereby realizing the so-called *place-and-play* concept [35], [36].

An underlying framework for self-managing devices is described in [37]: it includes measurement-based learning and adaptation to changing system context and application demands. [38] proposes a cognitive management framework for Smart Cities, where heterogeneous objects are represented in a virtualized environment, and cognition and proximity are used to automatically select the most relevant objects for the purpose of an application. However, neither work provides quantitative results for the proposed framework. Although the challenges that are to be addressed to enable self-awareness and self-configuration have already been identified, it is still not clear how to address them. In [36], a stack of solutions is built on top of the networking and service levels, creating a sort of Semantic IoT. The self-configurability of this framework is demonstrated by means of a demo application in the home automation domain. Reference [39] proposes an effective implementation to monitor regular domestic conditions through context awareness and learning tools.

Learning techniques should be adopted not only to adapt the context models when previously unseen data is encountered, but also to discover the relations between user contexts in the scope of the application requirements. To this aim, [40] proposes correlation mining algorithms based on Kullback-Leibler divergence and frequent set mining which exploit correlated contexts to enable unsupervised self-learning.

Although this is a lively research area, many issues still need to be addressed. One of the main concerns is about data mining: learning can happen only through experience, and billions of data points are needed to build effective learning schemes, train classifiers, etc., and this calls for the develop-

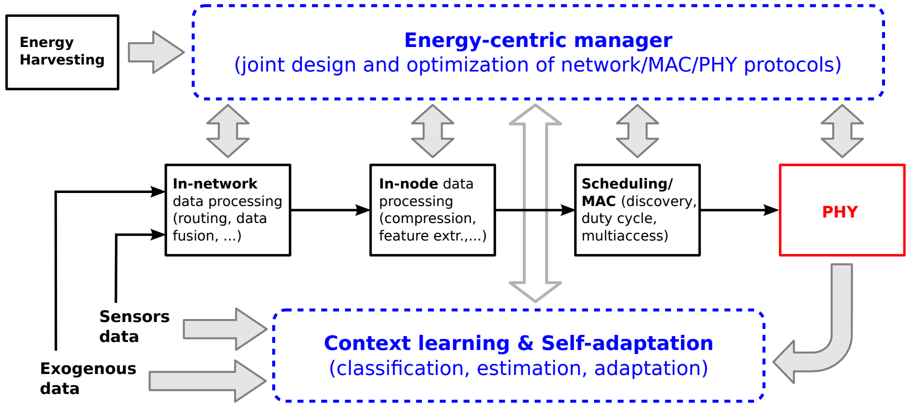

Fig. 1. Energy and context centric framework, showing the processing and communication blocks within each IoT device.

ment of efficient techniques for massive data collection [41]. Testbeds seem to fail due to their localized nature, to the often limited amount of data available and to the fact that it is hardly generated by real users. One concrete opportunity is offered by mobility and smart city data that start to be massively collected within real applications such as traffic and pollution monitoring.

Novelty: our approach is innovative in several respects. We adopt a modular design in the interest of an affordable complexity and we build signal processing and protocol elements around the data we measure. The approach is data-driven as its type, statistics and intensity (traffic rate) are used to understand what type of in-node processing serves best the application. For example, as we quantify below, lossy compression may be used to save transmission resources while still meeting certain (application-dependent) accuracy goals. While this is common practice, we advocate the on the fly estimation of rate distortion curves, without needing any a priori knowledge of which type of data the sensors provide. This, in turn, makes it possible to pick the most appropriate compressor and tune it to the desired operating point. In addition, processing figures (energy cost, representation accuracy, etc.) are fed to the network protocols (channel access, routing, sleep modes, etc.) to modulate their operating conditions and possibly change the type of protocol in use. Moreover, the protocols may as well induce changes in the signal processing algorithms to balance the amount of resources they allocate to different flows. So, the system operates in loops, where processing feeds protocols and vice-versa.

III. Energy and Context Centric System Framework

The reference framework we envision is sketched in Fig. 1. The solid boxes represent the functional modules that should be provided in peripheral IoT devices. The dashed boxes are logic modules, which can be either implemented in a centralized entity (e.g., the gateway), or distributed among the nodes in the form of look-up tables or if-then processes. Thin black arrows denote the information flow, whereas thicker arrows represent logical relations. Note that the actual implementation of the conceptual system framework sketched in Fig. 1 will depend on the device capabilities in terms of memory, energy and computation, and not all the processing functionalities have to be necessarily supported by each IoT device.

The role and purpose of each module are described below in greater detail.

Energy-Centric Manager (ECM). This logical block is at the core of the proposed optimization framework. Its objective is to improve the overall energy efficiency of the node and of the entire network while meeting certain application-dependent Quality of Service (QoS) requirements in terms of reliability, delay, and throughput. This is achieved through the joint optimization of network, scheduling, MAC, and PHY operations performed by the main functional modules, also accounting for the characteristics of the Energy Harvesting sources, if available. Accordingly, the ECM operates within a single node and across multiple nodes as we now explain.

At the intra-node level, the ECM acts on the in-node data processing and Scheduling/MAC blocks, taking into account Energy Harvesting sources. Its ultimate goal is to make an efficient use of the available energy resources, thus delaying or avoiding any battery depletion event, while maintaining the required QoS of data communication. To achieve this, the ECM needs to intelligently balance different technical aspects. For example, in-node data processing can be used to reduce the amount of data to be sent over the wireless interface, thus saving some transmission energy, though usually at the cost of losing part of the signal information, i.e., through lossy temporal compression or feature extraction of the locally sensed signals [42]. In addition, the energy drained by the data processing algorithms shall also be taken into account, to correctly evaluate the overall energy trade-off between compression and transmission. Finally, compression and transmission policies shall also consider the channel state as well as the intermittency of the energy source when EH is used, and accordingly schedule transmission events so as to avoid possible energy wastage [43], [44], [8]. By taking all these aspects into account, the ECM will determine the best combination (or a suitable approximation thereof) of in-node data processing (e.g., compression level) and transmission scheduling policies to provide the desired tradeoff between energy efficiency, communication reliability and throughput.

At the inter-node level, the ECM acts on the in-network data processing block with a network-wide perspective. Here, the aim is to find the proper combination of traffic filtering, packet routing (path selection), data aggregation (i.e., spatio-temporal compression) and data fusion algorithms. With filtering we aim at refining data flows into what the end user (or the application) actually needs. For example, a powerful node may be able to compute some relevant features of a flow and send those to the IoT gateway in place of the original data. Both filtering and aggregation are meant to avoid redundant transmissions over the physical links, exploiting the redundancy inherently present in the data. Data fusion basically solves a distributed estimation problem for a certain physical measurement or process through the distributed (and joint) processing of data from heterogeneous sources. The routing functionality shall be designed jointly with these aspects to facilitate aggregation and fusion opportunities, while meeting energy, computation, delay and throughput requirements. We remark that, although data aggregation is a mature research field [45], the way we look at it is novel. Specifically, we take inspiration from network function virtualization [46] in the Internet and advocate that similar concepts can be successfully applied to the considered IoT settings. That is, routing, filtering, aggregation and fusion functionalities shall be dynamically assigned to the nodes (and possibly rotated among them) according to their available computation, communication, memory and energy resources. We aim at distributed techniques to achieve this at runtime, while meeting the QoS constraints dictated by the application.

Energy Harvesting (EH). Energy harvesting functionalities enable the continuous collection of energy from the environment or from an external and controlled energy source, and hold the promise of energy neutral network operation [47]. EH technologies successfully exploited in the context of IoT include light (solar or indoor), thermal and vibration, and may widely differ in terms of usage and efficiency. The EH module influences every system module and processing functionality, e.g., scheduling and compression algorithms, and consequently is an integral part of the optimization framework. A realistic, but still useable, characterization of the inherent features of the EH process (e.g., intermittency of the energy source) is of vital importance, along with their performance implications, and the optimization of the system using tools such as learning [48] and sub-optimal approaches [49]. In addition to the scavenging of ambient energy, the framework also includes wireless energy transfer (WET) and cooperation techniques, where energy can be purposely transferred from an energy-rich node (e.g., a dedicated gateway) to other devices [50]. We foresee the possibility of employing WET to further boost the network performance [51], [52], in particular following the wireless powered communication network paradigm described in [53], [54], [55].

Context-Learning & Self-Adaptation (CLSA). This module is aimed at acquiring context information regarding, e.g., the type of signal(s) generated by the node's sensors, the nature of the cross-traffic coming from other nodes, and the nature and conditions of the available communication channels. This information is then used to identify the scenario of use and, then, adapt the optimization actions taken by the ECM. The idea is to come up with self-adapting policies, according to different contexts. Context-awareness [29], [56], [57] can be achieved by means of learning techniques [58], [59], by observing the type and format of the data flow generated by each IoT device. A fundamental feature of the proposed framework is the capability of offloading processing and protocol functionalities to the most energy-rich and computation-capable devices, thereby making this vision practically implementable on realistic IoT platforms. For example, intensive learning tasks could be carried out by the IoT Gateway, which could in turn offload some of the computational burden to Internet Cloud servers if needed. Conversely, energy constrained IoT nodes will only have to execute basic (possibly pre-computed) policies or to perform a limited number of operations. This approach will be pursued by additionally focusing on the decentralization of learning, routing and processing functionalities to those IoT nodes possessing more energy and computation capabilities.

Scheduling/MAC. The scheduling/MAC module manages the data transmission events and tunes the transmission parameters according to the expected channel and interference conditions and to the energy perspective of the device [43], [44], [60], i.e., the current energy reserve, the probability of gathering new energy from ambient sources or via WET, and the energy cost of compression algorithms, which will be dynamically and jointly managed with channel scheduling decisions [61], [62]. The MAC protocol shall be designed to exploit the possible advanced (e.g., multi-packet) reception capabilities of the receiver [22], radio duty cycling and the presence of a wake-up radio, if available. Also, self-adaptation to traffic type and channel asymmetries (downlink vs uplink), as well as different Tx/Rx capabilities (e.g., directional operation) are key considerations. For example, Time Division Multiple Access (TDMA) could be adopted under heavy traffic, whereas random access may be a better option when traffic is sporadic. The CLSA module will be utilized to this end.

A. Scenarios of interest

The IoT includes a very wide range of scenarios with different requirements and specifications. Here, we focus on some reference use cases that, despite their simplicity and level of abstraction, are representative of a broad range of applications.

A) Many to one – energy "rich": this first scenario concerns a single-hop network, where many nodes report data to a common receiver, which is (often) neither energy nor resource constrained. The sensor nodes may be either connected to the energy grid or battery-powered and, in the latter case, they could have energy harvesting capabilities. They can also perform some computation, although simple. In this context, it makes sense to assume that most of the signals generated by the nodes are time series as they are involved in the monitoring of some spatio-temporal physical process. Common applications for this scenario can be found in smart cities, e.g., traffic monitoring and environmental applications (noise level, humidity, vibration, light intensity and infrared, etc.). This architecture may also apply to assisted living and smart building scenarios, e.g., for indoor activity tracking.

B) Many to one –"smart dust": we consider a single-hop network, but we distinguish between two categories of nodes. *Type 1* sensors have very little energy available and form the so-called "smart-dust": they can be densely deployed, harvest energy from the environment and use it to transmit "once in a while". The scarce available energy does not allow them to perform heavy computation. The data sent by these devices is collected by *type 2* nodes, which have the same capabilities as those in scenario A), and communicate towards a powerful common receiver. In this case, the aggregated capabilities of *type 1* devices are much larger than that of a single *type 1* node, and processing techniques and scheduling protocols should leverage on this. This architecture may model, e.g., smart farming applications, where a lot of sensors with low computational capabilities are spread in the fields to measure acidity, soil moisture and temperature, etc., but also smart logistics operations where goods are to be tracked or checked for integrity, etc.

C) Multi-hop networks: in this case the IoT nodes may not be able to communicate directly with the gateway, and thus efficient routing protocols, that take into account the available energy and computational resources, are needed. A common application for this architecture is structural health monitoring, where sensors are embedded in a building or infrastructure (e.g., a bridge [63]) and monitor its status. Another case may be in smart cities, where sensors are directly connected with a gateway, but self-starting multi-hop routing is required to grant connectivity in the cases of gateway failures or poor channel quality (due to obstacles and interference). Although multi-hop connectivity has been around for years, and proven to work in some practical systems, it still represents a challenge from an energy point of view, because of the non negligible overhead required to maintain multi-hop routes in dynamic environments and/or the high packet-forwarding cost incurred by those nodes close to the sink. Under this premise, we will limit our interest to connectivity that spans only a few hops (up to three), which does not require complex routing algorithms and can instead exploit the presence of a limited number of relay nodes to alleviate the burden of packet forwarding.

IV. Reference scenarios and assumptions

In the following, we consider a channel access example which elucidates some aspects of the proposed framework. In this section, we will first introduce the considered mathematical models for the in-node data processing, energy consumption of the devices, and communication channel (Sections IV-A–IV-C). Then we will delve into data mining and classification algorithms in Section V-A, and, finally, we will use these results to study a channel access problem where multiple sources transmit heterogeneous data to a gateway (Sections V-B–V-D). This example refers to scenario A, where nodes possess some computation capabilities, that are exploited to compress the source signals.

A. In-node processing

In the above scenario A, IoT nodes are capable of processing data. One possible way to exploit their computation capabilities is to apply lossy compression to the time series they sense from the environment. This makes it possible to trade some accuracy in the data representation for an increased energy efficiency of the data transmission to the gateway. Note that to ensure a bounded reconstruction error at the receiver, while using the appropriate compression level, a reasonably accurate rate-distortion function for the sensed signals is required. Rate and distortion are formally defined as follows.

Definition 1 (Rate): Given a time series $\mathbf{x}$ and its compressed representation $\mathbf{y}$, we define the compression rate as:

$$\eta_c = \frac{N_b(\mathbf{y})}{N_b(\mathbf{x})} , \tag{1}$$

where $N_b(\mathbf{x})$ and $N_b(\mathbf{y})$ are the number of bits required to represent the original time series $\mathbf{x}$ and the compressed signal $\mathbf{y}$, respectively.

Definition 2 (Distortion): Given a time series $\mathbf{x}$ and its reconstructed version $\hat{\mathbf{x}}$, we define the distortion over N time samples as:

$$\hat{\varepsilon} = \frac{\max_{i=1,\ldots,N}\{|x_i - \hat{x}_i|\}}{\max_i\{x_i\} - \min_i\{x_i\}} \cdot 100 , \tag{2}$$

which corresponds to the maximum distance between the samples of $\mathbf{x}$ and $\hat{\mathbf{x}}$, normalized to the range of the values.

Rate-distortion curves are signal- and algorithm-dependent. For any given compression method, they can be empirically obtained by applying the algorithm to a given signal using different levels of compression (rate) and measuring the corresponding reconstruction error (distortion). The set of rate-distortion pairs can be then fit using a suitable function. In this paper we use the following model, whose shape resembles that of the rate-distortion curve of a Gaussian source [64]:

$$D = b\left(\frac{1}{\eta_c^{\alpha}} - 1\right) , \tag{3}$$

where $\alpha, b > 0$. Once α and b are known, (3) permits to gauge the distortion for any rate, i.e., any level of compression. This knowledge, depending on the specific application, can be exploited at the end nodes or at any intermediate point acting as a centralized manager that optimizes the network protocols. In Fig. 2, we show empirical rate-distortion points for three signal classes along with the corresponding fitting curves, obtained adapting Eq. (3).

B. Energy and power consumption

To design energy efficient algorithms and protocols it is key to identify and characterize all the sources of energy consumption and supply. It is hard to define an exhaustive and general model for the energy dynamics of an IoT device, since its energy consumption highly depends on the technology it employs, its operating conditions, and the algorithms it

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2692522, IEEE Access

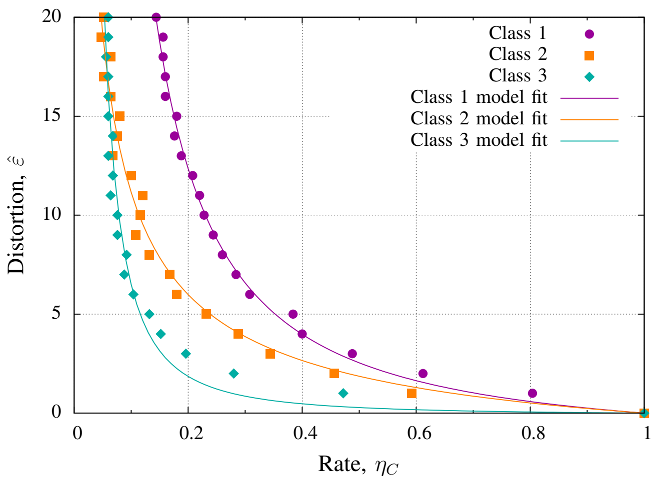

Fig. 2. Empirical fits for the rate-distortion Eq. (3) for three signal classes.

uses. Next, we describe a parameterized model that tries to capture all the major sources of energy expenditure, namely, communication, data acquisition, processing, and circuitry.

Sensing. Let N_s be the number of sensing events performed in a given time window T_s. The sensing energy is defined as:

$$E_{\mathrm{s}} = N_s \cdot E_{\mathrm{sens}} \tag{4}$$

where E_{sens} is the energy spent by the device to collect one sample. For periodic sensing, $N_s \simeq \mathrm{round}(T_s/T_p)$, where T_p is the nominal sensing period, while for aperiodic sensing, where the acquisition of samples is triggered by some event, N_s is a random variable whose distribution depends on the specific sensing process and on the observation window T_s.

Often, E_{sens} is very small compared to the energy drained by the RF architecture, and E_{s} becomes negligible. However, there exist devices such as cameras that may spend non negligible amounts of energy to collect a new image every few tens of milliseconds.

Data processing. In our example, we only focus on in-node compression operations, whose energy cost can be quantified using the results in [6], and define the energy consumption due to processing as:

$$E_{\mathrm{p}} = E_0 \cdot L_0 \cdot N_{\mathrm{c}}(\eta_{\mathrm{c}}), \tag{5}$$

where E_0 is the energy consumption per CPU cycle (that depends on the micro-controller unit), L_0 is the number of bits used to represent the original signal, and $N_{\mathrm{c}}(\eta_{\mathrm{c}})$ represents the number of clock cycles per bit needed to compress the input signal and is a function of the compression ratio η_{c}. Note that $N_{\mathrm{c}}(\eta_{\mathrm{c}})$ depends on the compression algorithm.

For what concerns channel coding, typically the energy it requires is assumed to be negligible with respect to the overall energy consumption and only the energy needed by the receiver for decoding is taken into account [65], hence we consider this contribution only in terms of variation of the number of bits to be transmitted over the air (i.e., redundancy bits added for FEC/CRC).

Transmission. The energy cost of any wireless transmission period can be modeled as:

$$E_{\mathrm{tx}} = \frac{\tau \cdot P_{\mathrm{tx}}}{\eta_{\mathrm{A}}}, \tag{6}$$

where τ is the transmission duration, P_{tx} is the average radiated power, and $\eta_{\mathrm{A}} \in (0, 1]$ is a constant that models the efficiency of the antenna's power amplifier. This source of energy consumption should be considered for all transmissions performed by the IoT device, and thus also includes retransmission attempts and all control messages, e.g., related to scheduling maintenance/generation of the access schedule in coordinated access schemes.

Reception. When receiving a packet, the device spends energy to receive the radio signal, which can be modeled analogously to Eq. (6), and to reconstruct the original data from its compressed/encoded version. This latter contribution is highly algorithm-dependent and, to the best of our knowledge, there exists no general expression to characterize it. Also the energy required by advanced decoding algorithms (e.g., interference cancellation) should be taken into account. However, in the example application that follows we mainly focus on the energy consumed for transmission, because the applications we target (scenario A of Section III-A) assume single-hop networks where the data sink is an energy-rich device.

Circuitry. We also consider the "basal" energy spent by the circuit in each of the node's possible operating states, $x \in \{\mathrm{sleep}, \mathrm{idle}, \mathrm{active}\}$. A simple way to model it is the following:

$$E_{\mathrm{c}} = T_x \cdot \varepsilon_{\mathrm{c},x}, \tag{7}$$

where $\varepsilon_{\mathrm{c},x}$ is the rate of circuitry energy consumption when the node is in mode x, and T_x is the time spent by the device in that mode. Also, going from mode x_1 to mode x_2 consumes energy, which is modeled as a constant contribution only depending on the two modes:

$$E_{\mathrm{switch}} = k_{x_1,x_2}, \tag{8}$$

The switching time is assumed to be negligible, and for this reason E_{switch} does not depend upon it.

C. Channel model

We consider communication channels that are independent among users and affected by path loss and block fading (e.g., Rayleigh fading). Since we are focusing on Scenario A, it is meaningful to assume that Channel State Information (CSI) is available at the nodes, which can exploit it to, e.g., perform power control. If the energy available to an IoT node is sufficient to transmit, the packet can be sent to the receiver but may not get through because of bad channel conditions (or collisions, in case of random access). If the latency requirements are strict or the communication model does not account for retransmissions, the information contained in a corrupted or lost packet cannot be recovered.

V. From Theory to Practice: A Combined Learning and Resource Scheduling Problem

In this section, we elaborate on the automatic classification of sensor signals. Our objective is to reliably predict the rate-distortion function of a generic temporal signal by analyzing a small window of samples.

A. Context classification

We foresee a usage model where data is gathered and, upon collecting a few samples (e.g., 500 samples are used for the results that follow), the time series is automatically classified in terms of rate-distortion behavior for a selected compression algorithm. Having this function, or at least a good estimate of it, makes it possible to decide upon the most suitable compression algorithm to use and to automatically tune it. In addition, besides compression at the source, the estimated rate-distortion tradeoff can be exploited to design and/or adapt the network protocols, which shall be jointly optimized with the compression algorithm, as we shall see in Section V-B. With respect to the energy-centric framework of Section III, this rate-distortion classifier falls within the "In-node data processing" and the "Contex Learning & Self-adaptation" blocks of Fig. 1.

For the purposes of this example, we collected diverse univariate real world time series, which were acquired from publicly available datasets. These have been selected as a representative set of the signal types for common IoT scenarios, including: 1) environmental sensing, e.g., temperature, humidity, soil moisture, precipitation measures, wind speed, solar radiation, 2) biomedical applications, e.g., electrocardiograms (ECG), photoplethysmograms and respiration signals, 3) smart electricity grids and smart cities, e.g., power consumption of home appliances and measures of buildings structural strain. In total, we have run experiments on 7010 time series taken from these application domains.

Every signal in the database is sampled at a constant rate which is signal-specific, but our aim is to come up with algorithms that are agnostic to it. For our experiments, we split each time series into non-overlapping temporal windows of N samples, so that each time window is an array of N real values, i.e., $\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^N$. With "input time series" we refer to one such window of data for a specific signal type. Based on preliminary results on compression schemes, and considering the analysis in [6], we group the signals into three classes, namely: i) *noise-like* signals, such as wind speed and structural strain, where the temporal correlation is low and the time series show an erratic behavior (difficult to predict and with no evident trend or periodic components); ii) *quasi-periodic* signals, such as ECG traces and other biomedical time series, where a similar pattern is repeated over time, with variations in shape and duration; and iii) *trend* signals, such as temperature, humidity, and other environmental quantities, which exhibit a slowly varying behavior and have a noticeable trend component. The signals in each class are expected to perform similarly when going through the process of (temporal) compression. Moreover, each time series, after being classified, can be associated with a certain rate-distortion curve, which is representative of the class it belongs to. The rationale is that this curve can be used to optimize the operation of networking protocols, e.g., to minimize the energy expenditure entailed by data collection algorithms, given an error tolerance for the signal reconstructed at the sink.

From the analysis in [6], we consider two lossy compression algorithms, which are suitable for IoT sensing-and-report applications, namely, i) Lightweight Temporal Compression (LTC) [7] and ii) compression based on Discrete Cosine Transform (DCT). LTC is among the most lightweight compression techniques for WSNs, whereas DCT-based algorithms usually provide the best accuracy, but are more energy demanding. Both compression schemes take as input the data to compress, $\mathbf{x}$, and an error parameter, ε, and output a model $\mathbf{y}$ for the compressed signal. The model is then transmitted and used at the sink to obtain the reconstructed signal $\hat{\mathbf{x}}$.

The classification procedure that we developed is based on the extraction of features from the original time series. In particular, it consists in a sequence of operations including: feature extraction, feature normalization, feature selection, and a final classification phase, which is carried out either using a Feed Forward Neural Network (FFNN) or a Support Vector Machine (SVM). A flow diagram of the proposed approach is shown in Fig. 3.

The feature extraction phase is performed through the Highly Comparative Time Series Analysis (HCTSA) framework of [11], which includes a large collection of methods for time series analysis and makes it possible to convert a time series into a vector of (thousands of) informative features, each obtained from a specific operation on the temporal signal. Each HCTSA operation is encoded as an algorithm taking as input a time series $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, and returning a single real number f_i, called a signal's *feature*. The collection of all the output features for an input time series is referred to as feature vector $\mathbf{f} = (f_1, f_2, \ldots, f_M) \in \mathbb{R}^M$. For our experiments, we considered a set of over 7000 time series of fixed length $N = 500$ samples, obtained from consecutive non overlapping portions of the three signal classes above. We have first run the feature extraction procedure on the input time series applying all the operations in the HCTSA library, which outputs 5254 features per time series. These are then normalized using an outlier robust sigmoidal transform and stored into a matrix $\hat{\mathbf{F}}$. The automatic classifiers are obtained by training SVMs and FFNNs using the features in $\hat{\mathbf{F}}$ and evaluating the classification accuracy for each signal class.

Fig. 4 shows that using either an SVM or an FFNN, the classification accuracy achieved using all the 5254 features is very high, i.e., higher than 99.8%. The use of the full feature set is however computationally demanding and impractical, especially if this classification task has to be carried out at the network edge (i.e., at the IoT nodes). We thus have to substantially reduce the number of relevant features to compute, in the hope that this will still lead to high classification rates. Driven by this, in Fig. 4 we also show the classification accuracy obtained when the SVM classifier is trained using the first L Principal Components (PCs) of $\hat{\mathbf{F}}$, with $1 \leq L \leq 10$, which grows from 73.43% using just the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2692522, IEEE Access

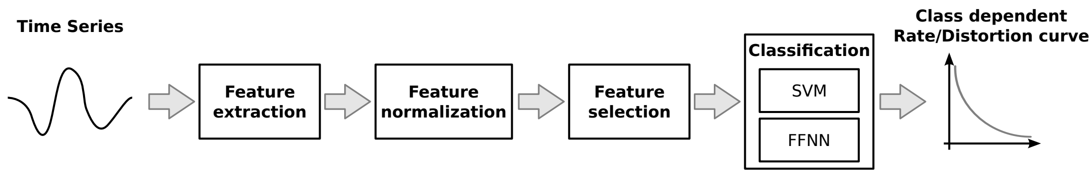

Fig. 3. Flow diagram of the proposed classification procedure.

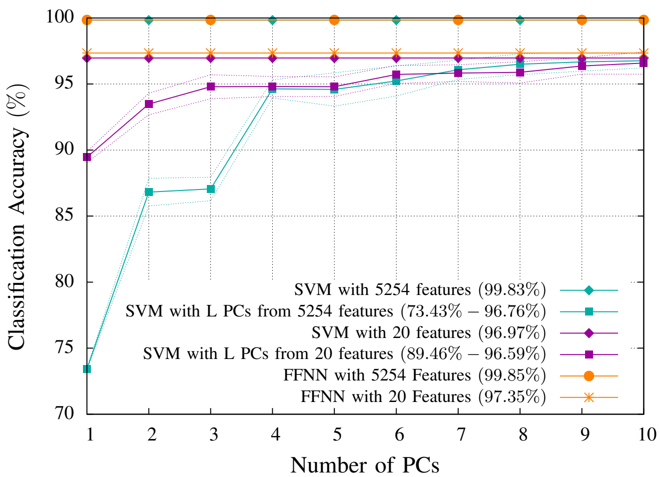

Fig. 4. Classification accuracy using a 10-fold cross validation approach for: 1) an SVM and an FFNN classifier trained on all the 5254 features extracted (i.e., on the entire matrix $\hat{\mathbf{F}}$), 2) an SVM classifier trained on the L principal components of $\hat{\mathbf{F}}$, 3) SVM and FFNN classifiers trained on the twenty most representative features, selected through a greedy procedure and 4) an SVM classifier trained on $L \leq 10$ principal components of the reduced and normalized $S \times 20$ signal-feature matrix.

first PC to 96.76% when using the first ten PCs. Computing feature vectors of ten elements (the first ten PCs) is certainly appealing, but this still entails the fact that the whole feature set has to be obtained first, which is still computationally prohibitive. For this reason, we applied a greedy feature-selection scheme to extract the twenty most representative features from the 5254 that were originally derived from the signals, obtaining a reduced and normalized $S \times 20$ signal-feature matrix, where S is the number of signal examples. This feature selection procedure is heuristic and better feature sets may be extracted through more involved (but computationally demanding) approaches. Nonetheless, it makes it possible to identify twenty pre-defined features from the original dataset, thereby considerably reducing the processing cost. We then trained SVM and FFNN classifiers using the so identified twenty features and the corresponding classification results are also shown in Fig. 4. As we can see from this plot, SVM and FFNN classifiers in this case still lead to good accuracies, about 97%.

Hence, upon training the classifier, each source can retrieve, with high accuracy and low computation cost, the rate-distortion function that best represents its signal. This only requires the inspection of small windows of data (500 samples in our tests) and the process can be repeated from time to time to track changes in the signal statistics. Classifiers shall be trained offline by a powerful node, but they are lightweight to execute and have a small memory footprint.

B. A channel access optimization problem

Our final goal is the definition of an agile MAC protocol, which dynamically tunes its parameters according to the evolution of the channel access scenario, by possibly following a principle of optimality, and switching between coordinated (e.g., TDMA-like) and random (e.g., Aloha-like) access schemes. The rate-distortion curves of Section IV-A are utilized to quantify the trade-off between distortion and energy consumption. Here, we present the main algorithms and results derived in [66], which solve a coordinated access optimization problem, and represent a first step towards the design of an agile MAC protocol.

We consider a centralized MAC scheduling, where a central entity (e.g., the IoT gateway) computes and disseminates the actions that every node performs, namely, the access schedule, the transmission powers and the source rates. The goal is to find a *policy* that simultaneously prolongs the network lifetime and satisfies some QoS requirements in terms of signal distortion at the receiver.[1] However, these are generally conflicting objectives, since lowering the signal distortion is generally possible through an increased energy consumption, which in turn impacts the network lifetime. In practice, there is a trade-off between the total amount of information transmitted and its quality. We also recall that in the following example we do not consider external renewable energy sources, thus the lifetime of the devices is always finite.

Next, we discuss how to find the policy that defines the MAC protocol, noting that our procedure is rather general and can also be employed in other settings. Formally, the optimization problem requires to explicitly assign the energy to be consumed at every time instant, taking into account the expected amount of data to transmit in the future, the future energy requirements, and other non-controllable factors (e.g., future channel conditions). We consider a slotted time system, comprising U IoT sources that send compressed signals to a common gateway at every time frame. Frames are divided into U slots in a TDMA fashion, and the slot durations are defined by the gateway and assigned at the beginning of each frame. In addition, within every slot, a node must decide the transmission power to use and the compression level for the signal it transmits. In practice, the slot duration and the transmission power influence the maximum number of bits that can be sent over the communication channel (e.g., using Shannon's

[1]For example, a practical QoS requirement may be to keep the signal distortion below a certain threshold for all the sources, with a fixed probability.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2692522, IEEE Access

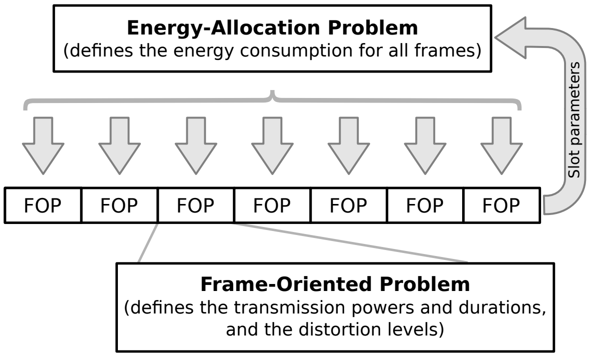

Fig. 5. Structure of the MAC layer optimization problem.

capacity formula, the maximum number of transmitted bits scales linearly with the duration and logarithmically with the power). This consequently imposes a bound on the signal distortion, which is intrinsically related to the length of the transmitted signal.

From our previous description, it becomes apparent that there are many variables to optimize (transmission powers, durations, and the compression levels for every source node). To address this problem, we decompose it into two connected sub-problems, as follows.

1) ***Energy-Allocation Problem (EAP).*** This is the main optimization process, which aims at allocating the energy that the sources consume in every transmission frame. In addition to the current frame, EAP should also consider what may happen in the future and the corresponding energy requirements.
2) ***Frame-Oriented Problem (FOP).*** Provided that EAP fully defines the energy to be used in each frame, the frame-oriented problem has two objectives: first, it finds the optimal slot allocation; second, it specifies the transmission powers and the compression levels for all nodes in order to make the best use of the allotted energy.

An illustrative example of how EAP and FOP interact with one another is provided in Fig. 5. The two problems are strictly connected, as the outcome of one block influences the choices of the other. However, they can be iteratively and independently optimized to solve the overarching optimization problem and find the desired trade-off between distortion and lifetime. The rationale behind this approach and the involved trade-offs are discussed in the rest of this section. More technical details can be found in [66].

C. Frame-Oriented Problem (FOP)

FOP optimizes the number of access slots, the transmission powers/durations and the distortion level for each source. The energy consumption is *fixed*, since all the energy assigned by EAP is meant to be fully used within each frame. Therefore, there is no lifetime issue in this phase. To more precisely define our objective, we introduce a network notion of distortion, that we call D_{net}. Since there are U sources transmitting their signals to a common receiver, it would be natural to consider the average distortion across all of them as our objective. This however does not ensure fairness among nodes and may lead to very unbalanced schedules. A more effective approach consists in defining the objective as the maximum level of distortion:

$$D_{\text{net}} \triangleq \max_{i=1,\ldots,U} D_i \tag{9}$$

Note that minimizing (9) amounts to enforcing a min-max utility optimization approach. This is particularly sensible in heterogeneous systems, where, otherwise, some nodes might be heavily penalized (e.g., because of the near-far effect, of their different traffic patterns or rate-distortion curves).

FOP solves the following optimization problem, minimizing D_{net}, subject to slot and energy constraints (the problem is fully specified and solved in [66]):

minimize:	D_{net}
variables:	tx durations
	tx powers
subject to:	TDMA-slot division
	energy imposed by EAP

Note that the most important constraints concern the TDMA structure of the frame (the number of time slots, which translates into a bound on the overall transmission duration of the nodes), and the maximum available energy dictated by EAP (this imposes constraints on transmission powers and durations). We also note that allocating the same slot duration to all nodes may be highly inefficient, as some of them may underuse the allocated resources.

Finally, we remark that the transmission power is a local parameter that has to be optimized according to: i) the efficiency of the transmission chain (e.g., the non-linearity of the power amplifier) and ii) the expected channel gain. Indeed, the frame-oriented problem implicitly depends on the state of the communication channels. In addition to the path loss coefficients, which represent the average channel quality, random fading should also be taken into account, as it may strongly influence the system behavior. In particular, in [66], both cases with and without full CSI are solved. When only statistical CSI is available, a probabilistic approach has to be employed to guarantee a sufficiently low distortion with a certain (positive) probability.

D. Energy-Allocation Problem (EAP)

EAP deals with the trade-off between network lifetime and signal distortion. The optimal working point generally depends on the application. For example, some scenarios may require very low or even zero distortion (e.g., the transmission of a binary information source, like an alarm), whereas others may accept lossy compression to a certain degree (e.g., the transmission of environmental signals). We model this trade-off as a multi-objective optimization problem, using a scalar λ to balance distortion and network lifetime. When $\lambda \to 1$, the network lifetime becomes the sole objective, whereas the distortion is the only objective for $\lambda \to 0$.

Furthermore, we generalize the notion of single-frame network distortion of Section V-C to multiple frames and we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2692522, IEEE Access

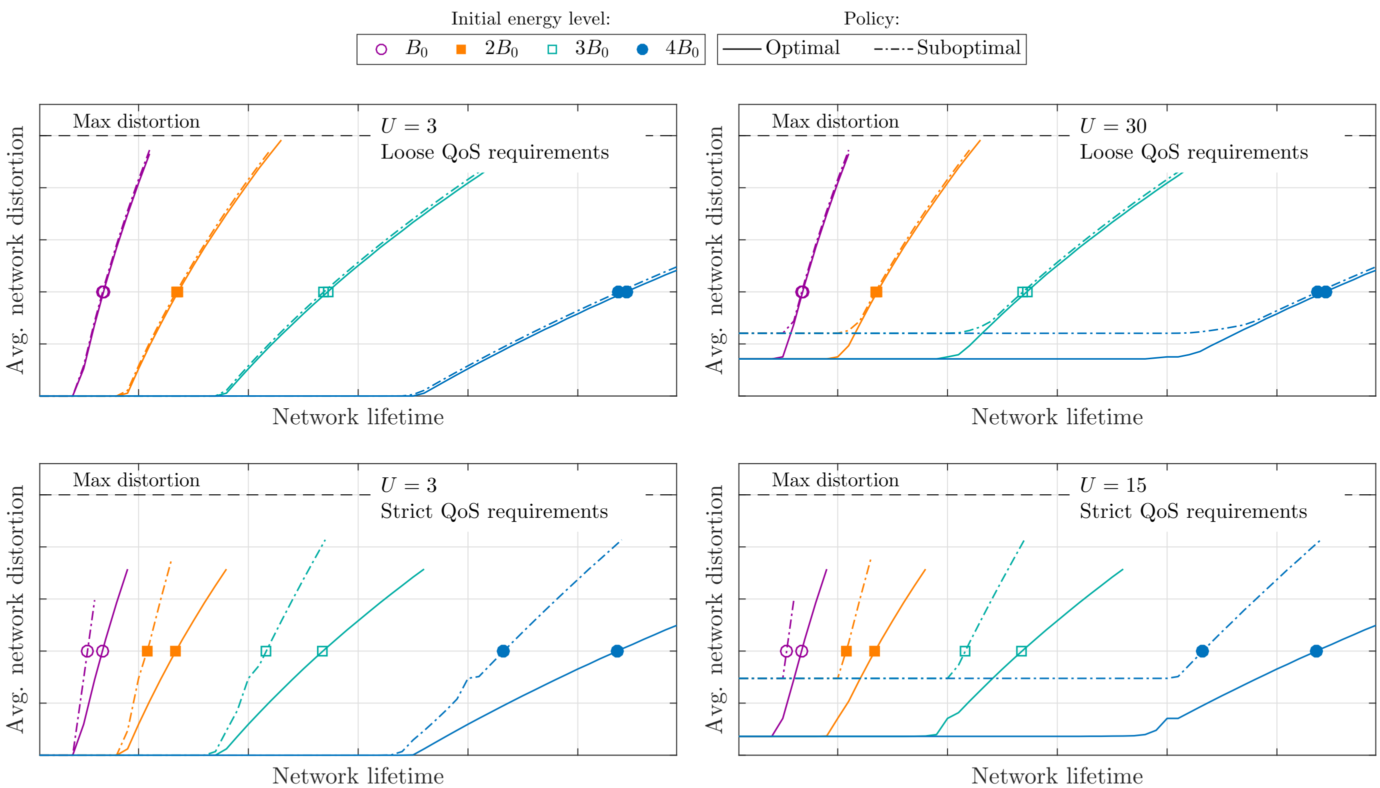

Fig. 6. Average network distortion as a function of the network lifetime using the optimal policy (solution of EAP and FOP) and a suboptimal approach for different numbers of users and QoS requirements. All curves are normalized with respect to an initial battery level B_0.

do so by taking the average of the network distortions as our optimization goal, i.e., the average over multiple frames of the maximum signal distortion in every frame. To solve EAP, we decompose the average distortion problem into U sub-problems, one for every source node. Then, a solution is found through a random alternate optimization algorithm, which optimizes the energy of one node at a time. In practice, EAP and FOP are tightly coupled and a single iteration of EAP requires to solve FOP multiple times.

E. Numerical results

To understand how the various parameters influence the system performance, we show an example network composed of three groups of nodes G_1, G_2 and G_3, placed at different distances from the gateway, with different rate-distortion curves, and with different QoS requirements. The transmission parameters are taken from the datasheets of two real devices, namely, the RN-131C 802.11 b/g Wireless LAN Module and the RC2400HP RF Transceiver Module. The parameters of the rate-distortion curves of Eq. (3) are derived according to the results of Section V-A.

In Fig. 6, we show the signal distortion curves as a function of the network lifetime, obtained solving the MAC layer optimization problem. Because of random fading, the channel conditions may be bad, and small distortions may be achieved only using a lot of energy (e.g., by increasing the transmission power) or performing long transmissions. Therefore, to avoid wastage of resources, we do not allow a node to transmit its data if the channel coefficient is below a certain threshold, which depends on the QoS requirements of the nodes, and on the number of sources (we consider an equal number of nodes in the three groups). In this work, we intend the QoS in terms of data delivery probability. Accordingly, the QoS requirements are said to be loose/strict when the service can tollerate high/low packet dropping probability, respectively. Therefore, the value of the channel-gain threshold below which the transmission is not attempted is high (low) when the QoS requirement is loose (strict).

The continuous lines were obtained following the optimization approach described in the previous section, which involves solving EAP and FOP iteratively, until convergence. Instead, the dashed lines are found using a simpler policy, that does not directly take into account the uncertainty about the future states and, in turn, is suboptimal. Indeed, as the number of nodes U increases or the QoS requirements become stricter, the distortion obtained with the suboptimal policy (dashed lines) is much higher than the optimal one. This emphasizes the importance of using a proper optimization approach at the MAC layer of energy constrained networks.

Note that, in all cases, the distortion is an increasing function of the lifetime, as expected. Moreover, when the lifetime is short, the curves are constant because 1) either a zero distortion has already been reached (cases with $U = 3$), or 2) it is not possible to consume all the energy available in the batteries (e.g., because of the constraints imposed by the communication channel and of the limited frame duration). We observe that, in these regions where the distortion remains constant for increasing values of the network lifetime, the best

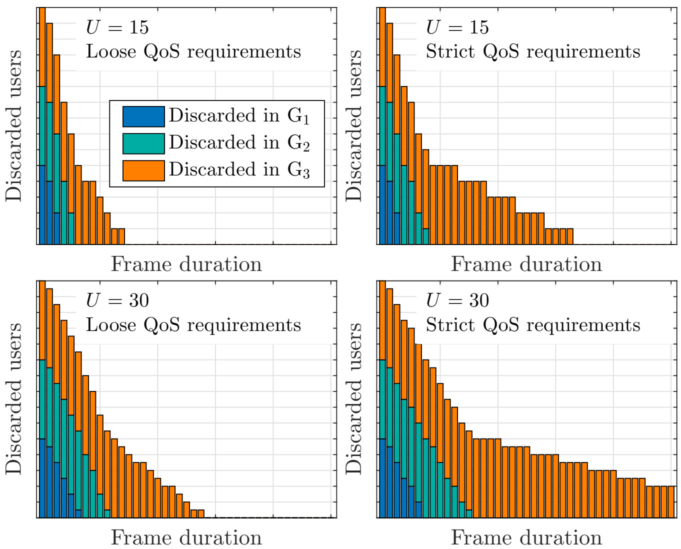

Fig. 7. Number of discarded users as a function of the frame duration for different numbers of users and QoS requirements.

operating point is clearly at the knee of the curve, where we get the longest lifetime for the same value of average network distortion. Moving beyond this point implies higher lifetimes at the cost of an increased distortion, and the operating point is then subject to application preferences.

A different effect can be observed in Fig. 7. In the x-axis we increase the frame duration, whereas in the y-axis we show the corresponding number of users that have to be discarded. Indeed, with short frames, it may not be possible to satisfy the QoS requirements of all nodes, especially when they are strict. Thus, some nodes are to be discarded and must wait for future transmission opportunities. Note that, the longer the frame, the smaller the number of discarded nodes.

VI. Conclusions and Lessons Learned

In this work, we have proposed a novel energy- and context-centric framework for the IoT, with the overall objective of prolonging the device energy subsistence, while guaranteeing a desired level of QoS. According to this framework, the protocol design effort should 1) target the reduction of the radio on-time (e.g., through data compression, smart radio duty-cycling, wake-up radios), 2) jointly manage scheduling (transmission) and higher layer processing (e.g., compression) to avoid energy wastage due to collisions, and 3) optimally tune the energy consumption in the presence of wireless channel impairments and intermittent energy sources (e.g., energy harvesting). Although the final objective is to build a comprehensive optimization framework, the large variety of possible application scenarios makes the specific formulation of the optimization problem context-dependent.

The results obtained in the initial development of such a framework have corroborated the effectiveness of the proposed approach. We focused on a scenario where multiple resource constrained sensors periodically report data to a common receiver, as common in monitoring applications, and can perform some (simple) computation. The in-node processing is data driven: the device dynamically decides upon the compression algorithm to use according to the type of signal it generates, whose properties may change over time. We used diverse real world time series to build a classifier based on a reduced set of features extracted from the signals themselves. We showed that good classification results can be obtained by using lightweight classifiers with humble storage requirements, such as pre-trained neural networks or support vector machines. The rate-distortion curves obtained through this process have been utilized in the design of the MAC layer through the definition of a scheduling problem that jointly considers channel and distortion requirements. We developed an optimization framework to optimize the performance (in terms of lifetime and/or average distortion at the receiver) of a network of heterogeneous devices. The numerical evaluation, based on realistic hardware parameters and signal models, shows that our approach can significantly outperform context-unaware systems, especially as the number of devices increases.

We believe that a resilient and effective IoT network can be realized only through the usage of context-aware protocols, that tune their settings and operating mode according to application requirements and to the dynamic conditions of the network and of the acquired signal. The energy dynamics of the devices also play a major role, and have a strong influence on all layers of the protocol stack. We are currently in the process of extending the framework through the addition of functionalities such as energy harvesting, protocol adaptation and wake-up radios.

Acknowledgment

This work was supported in part by Intel's Corporate Research Council thanks to the project "EC-CENTRIC: An energy- and context-centric optimization framework for IoT nodes."

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. ACM 1st Workshop on Mobile Cloud Computing (MCC)*, Aug. 2012, pp. 13–16.
[2] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.
[3] A. Bogliolo, V. Freschi, E. Lattanzi, A. L. Murphy, and U. Raza, "Towards a true energetically sustainable WSN: a case study with prediction-based data collection and a wake-up receiver," in *Proc. IEEE 9th Symp. on Industrial Embedded Systems (SIES)*, June 2014, pp. 21–28.
[4] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic, "Edge mining the internet of things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3816–3825, Oct. 2013.
[5] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "Practical data prediction for real-world wireless sensor networks," *IEEE Trans. on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2231–2244, Aug. 2015.
[6] D. Zordan, B. Martinez, I. Vilajosana, and M. Rossi, "On the performance of lossy compression schemes for energy constrained sensor networking," *ACM Trans. on Sensor Networks*, vol. 11, no. 1, pp. 15:1–15:34, Aug. 2014.
[7] T. Schoellhammer, E. Osterweil, B. Greenstein, M. Wimbrow, and D. Estrin, "Lightweight temporal compression of microclimate datasets," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*. IEEE Computer Society, 2004, pp. 516–524.
[8] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, compression, and recovery for wsns: Sparse signal modeling and monitoring framework," *IEEE Trans. on Wireless Communications*, vol. 11, no. 10, pp. 3447–3461, Oct. 2012.

[9] Y. Li and Y. Liang, "Temporal lossless and lossy compression in wireless sensor networks," *ACM Trans. on Sensor Networks*, vol. 12, no. 4, pp. 37:1–37:35, Oct. 2016.

[10] C. W. Tsai, C. F. Lai, M. C. Chiang, and L. T. Yang, "Data mining for internet of things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 77–97, First Quarter 2014.

[11] B. D. Fulcher and N. S. Jones, "Highly comparative feature-based time-series classification," *IEEE Trans. on Knowledge and Data Engineerin*, vol. 26, no. 12, pp. 3026–3037, Dec. 2014.

[12] F. Ganz, D. Puschmann, P. Barnaghi, and F. Carrez, "A practical evaluation of information processing and abstraction techniques for the internet of things," *IEEE Internet of Things Journal*, vol. 2, no. 4, pp. 340–354, Aug. 2015.

[13] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, Second Quarter 2010.

[14] W. Shen, T. Zhang, M. Gidlund, and F. Dobslaw, "SAS-TDMA: a source aware scheduling algorithm for real-time communication in industrial wireless sensor networks," *Wireless Networks*, vol. 19, no. 6, pp. 1155–1170, Aug. 2013.

[15] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TiSCH: deterministic IP-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, Dec. 2014.

[16] Y. Wu, X. Y. Li, Y. Li, and W. Lou, "Energy-efficient wake-up scheduling for data collection and aggregation," *IEEE Trans. on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 275–287, Feb. 2010.

[17] M. D. Jovanovic and G. L. Djordjevic, "Reduced-frame TDMA protocols for wireless sensor networks," *Int. Journal of Communication Systems*, vol. 27, no. 10, pp. 1857–1873, Oct. 2014.

[18] M. R. Lenka, A. R. Swain, and M. N. Sahoo, "Distributed slot scheduling algorithm for hybrid CSMA/TDMA MAC in wireless sensor networks," in *Proc. IEEE Conf. on Netw., Architecture and Storage (NAS)*, Aug. 2016.

[19] G. C. Madueño, Čedomir Stefanović, and P. Popovski, "Reliable and efficient access for alarm-initiated and regular M2M traffic in IEEE 802.11ah systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 673–682, Oct. 2016.

[20] A. Zanella, M. Zorzi, A. F. dos Santos, P. Popovski, N. Pratas, Č. Stefanovic, A. Dekorsy, B. Busropan, and T. A. H. J. Norp, "M2M massive wireless access: Challenges, research issues, and ways forward," in *Proc. IEEE Global Communications Conference Workshops (GC Wkshps)*, Dec. 2013, pp. 151–156.

[21] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, "Coded random access: applying codes on graphs to design random access protocols," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 144–150, June 2015.

[22] A. Zanella and M. Zorzi, "Theoretical analysis of the capture probability in wireless systems with multiple packet reception capabilities," *IEEE Trans. on Communications*, vol. 60, no. 4, pp. 1058–1071, Apr. 2012.

[23] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, "Low energy operation in WSNs: A survey of preamble sampling MAC protocols," *Computer Networks*, vol. 55, no. 15, pp. 3351–3363, Oct. 2011.

[24] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli, "Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems," in *Proc. IEEE Conf. on Computer Communications (INFOCOM)*, Apr. 2015, pp. 522–530.

[25] L. Gu and J. A. Stankovic, "Radio-triggered wake-up capability for sensor networks," in *Proc. IEEE 10th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, May 2004, pp. 27–36.

[26] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1172–1182, Sept. 2016.

[27] C. Perkins, E. Belding-Royer, S. Das *et al.*, "RFC 3561 - ad hoc on-demand distance vector (AODV) routing," *Internet RFCs*, pp. 1–38, 2003.

[28] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. on Wireless Communications*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[29] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, First quarter 2014.

[30] L.-H. Chang, T.-H. Lee, S.-J. Chen, and C.-Y. Liao, "Energy-efficient oriented routing algorithm in wireless sensor networks," in *Proc. IEEE Conf. on Systems, Man, and Cybernetics (SMC)*, Oct. 2013, pp. 3813–3818.

[31] J. V. V. Sobral, J. J. P. C. Rodrigues, K. Saleem, J. F. de Paz, and J. M. Corchado, "A composite routing metric for wireless sensor networks in AAL-IoT," in *Proc. IFIP 9th Wireless and Mobile Networking Conference (WMNC)*, July 2016, pp. 168–173.

[32] Y. Liu, Y. Yang, X. Lv, and L. Wang, "A self-learning sensor fault detection framework for industry monitoring IoT," *Mathematical problems in engineering*, vol. 2013, Aug. 2013.

[33] V. H. Bhide and S. Wagh, "i-learning IoT: An intelligent self learning system for home automation using IoT," in *Proc. IEEE Conf. on Communications and Signal Processing (ICCSP)*, Apr. 2015, pp. 1763–1767.

[34] A. Bogliolo, S. Delpriori, E. Lattanzi, and A. Seraghiti, "Self-adapting maximum flow routing for autonomous wireless sensor networks," *Cluster Computing*, vol. 14, no. 1, pp. 1–14, Mar. 2011.

[35] A. Biral, M. Centenaro, A. Zanella, L. Vangelista, and M. Zorzi, "The challenges of M2M massive access in wireless cellular networks," *Digital Communications and Networks*, vol. 1, no. 1, pp. 1–19, Feb. 2015.

[36] I. Chatzigiannakis, H. Hasemann, M. Karnstedt, O. Kleine, A. Kroller, M. Leggieri, D. Pfisterer, K. Romer, and C. Truong, "True self-configuration for the IoT," in *2012 3rd IEEE International Conference on the Internet of Things*, Oct. 2012, pp. 9–15.

[37] A. P. Athreya, B. DeBruhl, and P. Tague, "Designing for self-configuration and self-adaptation in the internet of things," in *Proc. IEEE 9th Conf. on Collaborative Computing: Networking, Applications and Worksharing*, Oct. 2013, pp. 585–592.

[38] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A. R. Biswas, and K. Moessner, "Enabling smart cities through a cognitive management framework for the internet of things," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 102–111, June 2013.

[39] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, Oct. 2013.

[40] A. K. Ramakrishnan, D. Preuveneers, and Y. Berbers, "Enabling self-learning in dynamic and open IoT environments," *Procedia Computer Science*, vol. 32, pp. 207–214, 2014.

[41] T. Palpanas, "Data series management: The next challenge," in *Proc. IEEE 32nd Conf. on Data Engineering Workshops (ICDEW)*, May 2016, pp. 196–199.

[42] D. Zordan, T. Melodia, and M. Rossi, "On the design of temporal compression strategies for energy harvesting sensor networks," *IEEE Trans. on Wireless Communications*, vol. 15, no. 2, pp. 1336–1352, Feb. 2016.

[43] T. Erseghe, A. Zanella, and C. Codemo, "Markov decision processes with threshold based piecewise linear optimal policies," *IEEE Wireless Communications Letters*, vol. 2, no. 4, pp. 459–462, Aug. 2013.

[44] N. Michelusi, K. Stamatiou, and M. Zorzi, "Transmission policies for energy harvesting sensors with time-correlated energy supply," *IEEE Trans. on Communications*, vol. 61, no. 7, pp. 2988–3001, July 2013.

[45] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: A survey," *IEEE Wireless Communication Magazine*, vol. 14, no. 2, pp. 70–87, Apr. 2007.

[46] European Telecommunications Standards Institute, "Network functions virtualisation - introductory white paper," in *SDN and OpenFlow World Congress*, Oct. 2012.

[47] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

[48] P. Blasco, D. Gunduz, and M. Dohler, "A learning theoretic approach to energy harvesting communication system optimization," *IEEE Trans. on Wireless Communications*, vol. 12, no. 4, pp. 1872–1882, Apr. 2013.

[49] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007.

[50] P. Grover and A. Sahai, "Shannon meets Tesla: wireless information and power transfer." in *Proc. IEEE Symp. on Information Theory Proceedings (ISIT)*, June 2010, pp. 2363–2367.

[51] L. Xiao, P. Wang, D. Niyato, D. Kim, and Z. Han, "Wireless networks with rf energy harvesting: A contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 757–789, Second quarter 2015.

[52] A. Biason and M. Zorzi, "Joint transmission and energy transfer policies for energy harvesting devices with finite batteries," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2626–2640, Dec. 2015.

[53] L. Liu, R. Zhang, and K.-C. Chua, "Multi-antenna wireless powered communication with energy beamforming," *IEEE Trans. on Communications*, vol. 62, no. 12, pp. 4349–4361, Dec. 2014.

[54] H. Ju and R. Zhang, "Throughput maximization in wireless powered communication networks," *IEEE Trans. on Wireless Communications*, vol. 13, no. 1, pp. 418–428, Jan. 2014.

[55] A. Biason and M. Zorzi, "Battery-powered devices in WPCNs," *IEEE Trans. on Communications*, vol. 65, no. 1, pp. 216–229, Oct. 2016.

[56] V. Cristea, C. Dobre, and F. Pop, *Context-Aware Environments for the Internet of Things*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 25–49. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-34952-2_2

[57] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1247–2376, Fourth quarter 2015.

[58] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[59] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge: MIT press, Sept. 1999, vol. 3, no. 9.

[60] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, "Designing intelligent energy harvesting communication systems," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 210–216, Jan, 2014.

[61] O. Ozel and S. Ulukus, "Achieving awgn capacity under stochastic energy harvesting," *IEEE Trans. on Information Theory*, vol. 58, no. 10, pp. 6471–6483, Oct. 2012.

[62] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1732–1743, Sept. 2011.

[63] S. Basagni, C. Petrioli, and D. Spenza, "CTP-WUR: The collection tree protocol in wake-up radio wsns for critical applications," in *Proc. IEEE Conf. on Computing, Networking and Communications (ICNC)*, Feb. 2016.

[64] T. Berger, *Rate-Distortion Theory*. John Wiley & Sons, Inc., Apr. 2003.

[65] S. L. Howard, C. Schlegel, and K. Iniewski, "Error control coding in low-power wireless sensor networks: When is ecc energy-efficient?" *EURASIP Journal on Wireless Communication Networks*, vol. 2006, no. 2, Apr. 2006.

[66] A. Biason, C. Pielli, A. Zanella, and M. Zorzi, "Energy/distortion tradeoffs in joint source coding and MAC scheduling for the IoT," *arXiv:1702.03695*, submitted to *IEEE Trans. on Wireless Communications*, Nov. 2016.