Computer Communications 160 (2020) 554–566

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

An optimal policy for joint compression and transmission control in delay-constrained energy harvesting IoT devices

Vesal Hakami [a,*], Seyedakbar Mostafavi [b], Nastooh Taheri Javan [c], Zahra Rashidi [a]

[a] School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[b] Department of Computer Engineering, Yazd University, Yazd, Iran
[c] Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Keywords:
Compression control
Constrained Markov decision process
Delay-constrained
Energy harvesting
Internet of Things
Reinforcement learning
Transmission control

ABSTRACT

Energy-efficient communication remains one of the key requirements of the Internet of Things (IoT) platforms. The concern on energy consumption can be mitigated by exploiting technical ploys to reduce the volume of data for transmission (e.g., via sensing data compression) as well as by resorting to technological advancements (e.g., energy harvesting). However, these mitigating measures carry their own cost, which is the additional complexity of control and optimization in the digital communication chain. In particular, compression ratio is another control knob that needs adjusting besides the usual transmission parameters. Also, with the random and sporadic nature of the harvested energy, the goal shifts from mere energy conservation to judicious consumption of the renewable energy in a foresighted manner. In this paper, we assume an energy-harvesting IoT device that is tasked with (loss-lessly) compressing and reporting delay-constrained sensing events to an IoT gateway over a time-varying wireless channel. We are interested in computing an optimal policy for joint compression and transmission control adaptive to the node's energy availability, transmission buffer length, as well as its wireless channel conditions. We cast the problem as a Constrained Markov Decision Process (CMDP), and propose a two-timescale model-free reinforcement learning (RL) algorithm that is able to shape the optimal control policy in the absence of the statistical knowledge of the underlying system dynamics. Exhaustive simulation experiments are conducted to investigate the convergence of the learning algorithm, to explore the impacts of different system parameters (such as: the rate of sensing events, the energy arrival rate, and battery capacity) on the performance of the proposed policy, as well as to compare against some baseline schemes.

1. Introduction

1.1. Research background

Internet of Things (IoT) as the main pillar of the fourth industrial revolution (Industry 4.0) prevails in today's Internet infrastructure. With the widespread use of IoT devices equipped with small batteries, lifetime maximization through energy optimization policies has attracted much attention in recent years. Energy conservation is even more important in mission-critical applications, where it is expected that the IoT device works for a long period of time without the need for battery or node replacement [1].

The energy management solutions in the literature are primarily geared towards lifetime maximization through effective use of the available energy capacity. In fact, due to the high cost of data transmission, in-network processing, involving operations such as filtering, compression, and fusion is a technique widely used in conventional wireless sensor and today's IoT for reducing the communication overhead [1–4]. Among the in-network processing techniques, compression is of particular interest given that it can be performed independently at every forwarding node; the downstream transmission rate of most stream-oriented data can be reduced by the application of appropriate compression algorithms, both lossless [5,6] and lossy [7–9]. However, it has been shown that applying maximum compression level is not an energy-efficient policy in most cases as it incurs additional delay and aggravates the computational complexity (e.g., [10,11]). Therefore, finding an efficient trade-off between compression and transmission is important to achieve higher energy savings without sacrificing much in terms of delay or other QoS criteria. Armed with this understanding, the problem of joint compression and transmission control (collectively, referred to as "communication control") has begun to receive attention due to its tunable complexity and delay costs [7,9,12,13].

* Corresponding author.
E-mail addresses: vhakami@iust.ac.ir (V. Hakami), a.mostafavi@yazd.ac.ir (S. Mostafavi), nastooh@aut.ac.ir (N.T. Javan), z_rashidi96@comp.iust.ac.ir (Z. Rashidi).

https://doi.org/10.1016/j.comcom.2020.07.005
Received 1 January 2020; Received in revised form 4 June 2020; Accepted 2 July 2020
Available online 4 July 2020

Another efficient way to improve the lifetime of IoT systems is through the use of recent technological advancements that have enabled energy harvesting capabilities for wireless nodes. Energy harvesting, as a promising complement to the conventional battery power of IoT devices, enables the sensor nodes to absorb energy from the environment, operating over a longer time horizon. In fact, exploiting the energy harvesting capability helps the IoT nodes to power themselves from theoretically unlimited energy sources that are present in their surrounding environment (e.g., in the form of solar, vibration, thermoelectricity, etc.).

Despite all the benefits associated with energy harvesting, the communication control problem becomes more complex in this context. This is due to the uncertainty associated with the battery charging process as the amount of harvested energy randomly changes over time. Also, by nature, wireless channel fading renders the link conditions stochastic and time-varying as well. To top it all off, one should also consider the uncertainty inherent in random sensory event generations. Hence, to account for the time-varying nature of these processes, a principled way to optimize the communication control policy of an IoT device is to capture these dynamics explicitly within a stochastic optimization framework with long-run objectives (e.g., expected cumulative energy consumption).

In this paper, we consider the problem of compressing and reporting sensory data packets for a single IoT device over a point-to-point link. The device is equipped with a rechargeable battery with energy harvesting capability. The random event data gathered from the environment passes through a lossless compression module before getting queued in a buffer for subsequent transmission. The wireless channel state, packet arrival quantities and the amount of energy harvested from the environment will vary randomly over time. We seek an efficient policy that adaptively tunes the lossless compression level as well as the transmission window (of packets) as the control knobs. Our objective is to minimize the average power (required for compressing and 'reliably' transmitting at a given rate), while satisfying a constraint on the average (buffering) delay of the event data.

In order to better highlight our contributions, we first give a brief account on the most relevant previous works, identify the research gap, and state our motivations.

1.2. Related work

A pioneer work which has raised the issue of tradeoff between the energy costs of transmission and compression is [10]. There, the authors have argued that while the computation energy of data compression is negligible for simple applications (e.g., temperature sensing), for advanced applications with heavy data flow, including structural health monitoring, video surveillance, and image-based tracking, compression of complex data sets is envisioned to cost energy comparable with wireless communication. Hence, blindly applying maximum compression may lead to extra energy cost compared to transmitting the raw data.

Motivated by the above observation, several research works have investigated the concept of tunable compression that is capable of tuning the computation complexity of lossless data compression based on the energy availability (e.g., [5,6,14]). Such a concept is also well-supported by the reality of popular compression algorithms; for example, the gzip program supports up to ten levels of different compression ratio, with larger compression ratio resulting in longer compression time and hence higher energy cost [15,16].

Based on the availability of information at the transmitter, the body of literature on the joint optimization of compression/transmission in IoT/sensor devices can be categorized into three distinct groups: *offline*, *online* and *model-free* schemes.

In offline optimization, it is basically assumed that exact information about the time and amount of data, energy arrival as well as the wireless channel state is available acausally (i.e., before decision making) at the sender side. An online optimization framework, on the other hand, only utilizes statistical information about the data, energy arrival and the wireless channel, but causal information about the process realization. However, in many real-world scenarios, it is not possible to attain exact or even statistical information from the non-deterministic, time-varying processes of data gathering, energy harvesting, or channel fading. Hence, both online and offline optimization approaches fail to achieve optimal energy consumption policies. In these scenarios, a model-free approach can instead be adopted to learn an adaptive communication policy through real-time interactions and experiences with the environment.

- Within the category of offline optimization schemes, the work by Tavli et al. in [5] has come up with a linear programming (LP) formulation for joint dynamic data compression and flow balancing to maximize the lifetime of wireless sensor networks (WSNs). In [14], the LP formulation of [5] has been extended to jointly consider dynamic compression along with the stealth mode of operation for the sensors. In the context of wireless multimedia sensor networks, the work in [12] has exploited the convex optimization theory to derive the optimal tradeoff between transmission and compression with the objective of network lifetime maximization under the delay quality of service constraints. In [7], a multi-objective optimization problem has been formulated to select a rate–distortion working point to conduct lossy compression at each source node, while jointly assessing a routing path for the compressed information, under distortion and capacity constraints. The study in [17] has exploited the NUM framework [18] to jointly optimize the source data rate, the degree of stream compression, and the location of fusion operators. A more closely related work to ours is [19], where it considers the joint use of data compression and wireless transmission speed control for wearable devices to minimize the total energy consumption while satisfying a deadline constraint. However, being an offline scheme, the solution given in [19] assumes an unrealistic setting where future channel gains are known ahead.
- As for online schemes, in [20], a solar-powered WSN is envisaged, and a simple threshold-based scheme is presented to decide whether there is any surplus energy. Nodes with residual energy less than a certain threshold transfer data with compression in order to reduce energy consumption, and nodes with residual energy over the threshold (which means there is surplus energy) transfer data without compression to reduce the delay time between nodes by using the surplus energy. The problem of online tradeoff between compression and transmission energy consumption is addressed more systematically in [21]. In particular, a formulation based on the formalism of Markov decision process (MDP) [22] is given in [21] for an energy harvesting WSN, with the objective of minimizing the average distortion of the compressed data in the long run under the energy variations. Another online scheme for joint data compression and wireless transmission speed control has been proposed in [19]. While the authors do not assume energy harvesting capability for the nodes, the future channel gains are assumed to be unknown and change stochastically. They show that their online algorithm, despite not knowing future channel conditions, closely approximates the performance of the offline optimal. Finally, Castiglione et al. has proposed an energy management policy in [23] for energy harvesting WSNs in which the energy management unit allocates energy to different units based on the statistics of energy harvesting, sensed data quality, signal-to-noise ratio and the size of data queue. In this study, the purpose of lossy compression is to make an optimal balance among the level of data distortion, the stability of the queue and the transmission delay. The proposed policy is adaptive to the energy status, channel, and flow of data generation.

- Model-free methods can further be divided into two subcategories: those based on Lyapunov optimization [24] and others that utilize learning-theoretic schemes, e.g., reinforcement learning (RL) [25] techniques. Probably one of the first studies that have utilized the Lyapunov optimization framework to address the dynamic compression problem is [6]. The setting considered in [6] consists of a multi-sensor device (with a conventional energy source) sending losslessly compressed data over a point-to-point fading channel. The baseline scheme in [6] has then been extended to a lossy compression scenario with distortion constraints in [8], and also to multi-hop communications in [26]. More recently, by adopting the Lyapunov framework in [27], a joint compression–transmission approach has been proposed for wearable devices which not only minimizes the energy consumption of both compression and transmission, but also maintains the corresponding data distortion and transmission delay within a certain tolerant level. The proposed scheme in [27], however, does not assume harvesting capability for the devices. Finally, there is the work done by Tapparello et al. in [28] for the sensor nodes capable of energy harvesting. They have come up with a dynamic compression/transmission strategy which is adaptive to the harvested energy, channel state, data queue, battery, and the source correlation statistical model.
  While Lyapunov optimization does not rely on the statistical knowledge of the system's stochastic dynamics to compute the optimal communication policy, it is only applicable to settings with more restricted stochastic assumptions. For example, standard Lyapunov schemes work by minimizing an instantaneous Lyapunov drift in each slot, while basically assuming that the underlying processes behind the channel variations, energy charging, and event generation are i.i.d. Moreover, the derived policy (e.g., dynamic backpresssure algorithm) by the Lyapunov drift theory and the Lyapunov optimization theory may not have good delay performance in moderate and light traffic loading regimes. It only allows potentially simple solutions with throughput optimality, which is a weak form of delay performance [29]. A more generalized method with the capability of handling delay-constrained communication control and of dealing with other types of stochastic processes is Makov decision problem [22] and RL [25].
  The work in [30] adopts the constrained Markov decision process (CMDP) formalism [31] to address lossy data compression for wireless transmission over fading channels in the presence of a stochastic energy input process and a replenishable energy buffer. An RL-based algorithm has then been designed to derive an optimal compression policy through a Lagrangian relaxation approach combined with a dichotomic search for the Lagrangian multiplier. The authors only consider compression control, there is no data buffer and hence, no guarantee on average delay performance. In fact, the objective function in [30] concerns data fidelity with a constraint on average energy consumption. A more mature RL-based scheme is [9], which tackles the problem of joint compression, channel coding and retransmission for an energy-harvesting-based multi-sensor monitoring system. The compression scheme considered in [9] is a lossy type, and they aim for minimizing the long-term average distortion at the receiver. The only control knob considered in [9] is power, and again, there is no guarantee on the average delay performance.

1.3. Research gap and motivation

According to our review of the related work in Section 1.2, the problem addressed in this paper is novel in the following respects:

- The majority of the studies on joint optimization of compression/transmission in WSN or IoT systems lie within the offline optimization framework (e.g., [2,5,7,12,14,17], and [19]), where it is unrealistically assumed that non-causal information regarding the exact trace of system states (i.e., channel, data, energy, etc.) is available beforehand. Also, there exist many online optimization schemes which more realistically assume that system states realize at run-time, but still, they require an explicit knowledge of the statistics of the system processes [20,21], and [23]. Our work in this paper differs from both these lines of work, as we approach the communication control problem from a model-free perspective.
- Compared to previous model-free schemes, on the one hand, we have those based on Lyapunov optimization (e.g., [6,8,26,27], and [28]). Based on our review of these works, there are at least two main fronts that form our motivation: first, there is no prior work that addresses a setting featuring lossless compression, energy harvesting, and delay-constrained communication. Moreover, as mentioned earlier, standard Lyapunov schemes are not suitable for more realistic stochastic dynamics with temporal dependency between correlated fading channel conditions, energy arrivals, and sensory event generation. This warrants a formulation based on a more generalized formalism of MDPs. On the other hand, we have the RL-based methods in [30] and [9]. None of these two schemes has addressed delay-constrained communication and lossless compression. In this paper, we consider more realistically the case of random event generation where event packets arrive randomly at different times. In such cases, the data arrival process also contributes to the dynamics of the system, which calls for a control policy adaptive to data queue state for handling the time-varying queueing delay experienced by the packets. Under these dynamics, the system's objective also needs to be constrained with an expected delay performance.

1.4. Contributions

Our contributions in this paper can be summarized as follows:

- We use the CMDP formalism [31] to formulate the joint lossless compression and transmission optimization problem in an energy harvesting IoT (sensing) device. Our formulation accounts for the time-varying channel, random event generation as well as the stochastic energy arrivals. Our objective is to minimize the discounted sum of energy consumption, subject to a specified delay constraint on the average data buffer waiting time. To handle to the constraint on average delay performance, we apply the Lagrangian technique to express the Bellman equations underlying the CMDP problem in a standard unconstrained form. This effectively paves the way to solve the original constrained problem by solving instead two (coupled) optimization problems, one in the space of control policies and the other in the space of Lagrange multipliers.
- We propose a model-free reinforcement learning algorithm that can compute the optimal solution pair (i.e., the control policy and Lagrange multiplier) in the absence of the statistical knowledge of the system, and instead, by relying only on the immediate feedbacks acquired through real-time interactions with the operating environment. The proposed learning algorithm consists of two iterative procedures: Q-learning [32] for computing the control policy, and stochastic subgradient-ascent for computing the optimal Lagrange multiplier. Despite these two problems are coupled together by definition, we utilize the technique of timescale separation from the stochastic approximation theory [33] to allow for their simultaneous updating with no risk of divergence.
- We conduct numerical analyses and experimental studies to evaluate our proposed algorithm in terms of its convergence properties, and its behavior under different intensities for event generation as well as varying energy charging rates, and energy buffer sizes. We also show that the learning algorithm achieves significant energy savings compared to baseline schemes which only optimize either the transmission or the compression blocks.

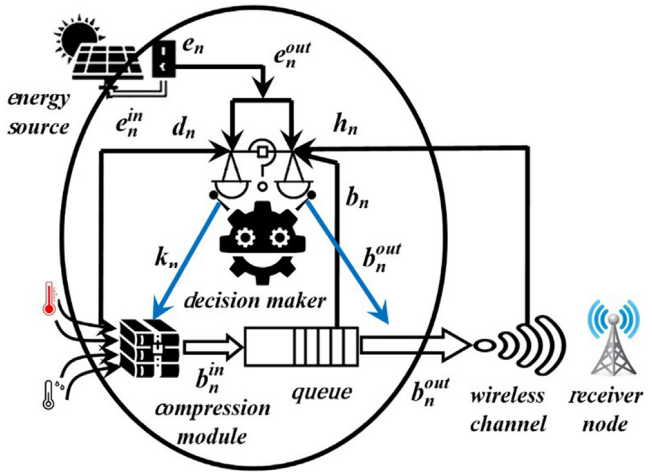

Fig. 1. Schematic of the IoT device.

1.5. Outline

The remainder of the paper is organized as follows: in Section 2, the system model and the assumptions are presented. In Section 3, we come up with the problem formulation in terms of a CMDP. Section 4 proposes a two-timescale reinforcement learning algorithm to find the optimal control policy. In Section 5, we evaluate the performance of the proposed scheme. The paper concludes in Section 6, where we also highlight some directions for future works.

2. System model and assumptions

In this section, we first introduce the system model for a delay-constrained energy harvesting IoT device. Then, we elaborate on the assumptions made about the dynamics of the energy charging process, sensory data generation, and the variations of the wireless channel state.

2.1. IoT device model

We consider a scenario in which a wireless IoT node is equipped with multiple sensors, and is tasked with sensing and reporting events from the environment (see Fig. 1). The IoT device is assumed to subsist on a rechargeable battery and energy harvesting circuitry, with no external power supply. In order to judiciously consume the limited harvested energy, the node has to intelligently decide on two things: First, the level of compression by which the sensed data can be losslessly compressed before insertion into the transmission buffer. Second, the number of units of data (i.e., packets) that are to be sent out over the wireless channel to the IoT gateway. For short, we also jointly refer to these two decisions as "communication control decisions".

2.2. Sensory event data generation model

We assume that the system operates in discrete time, i.e., the time is slotted and indexed by $n \in \{0, 1, 2, \dots\}$. In every timeslot, the device receives a random number of data packets from the sensors installed on it. The state space of the number of received data packets is represented as a discrete and finite space $\mathcal{D} = \{0, 1, 2, \dots, D\}$. The number of input data packets at time n is denoted by $d_n \in \mathcal{D}$.

Assumption 1 (*Data Generation Model*)**.** The dynamics of the sensor data generation follows a Markov chain. Note that a simpler (yet more common) assumption on the arrival process would be the special case where the process $\{d_n\}_{n\in\mathbb{N}}$ is i.i.d. ■

2.3. IoT device compression control module

We assume that packets arriving on the same timeslot contain correlated data, and that this data can be compressed using one of multiple compression options. However, the signal processing required for compression consumes a significant amount of energy, and more sophisticated compression algorithms are also more energy expensive. Similar to [22], the functionality of the compression module is described by a lossless compression function Ψ which has the compression capability at K different levels from set $\mathcal{K} = \{1, 2, \dots, K\}$ where level 1 means no compression and level K means compression at the highest ratio. In each timeslot, the compression function $\Psi(d_n, k_n)$ receives two parameters, i.e., the number of input data packets d_n, and the compression level $k_n \in \mathcal{K}$ (which is selected by the decision unit). Then, the compression function generates a random number b_n^{in} of data packets, depending on the correlation among the data and their compressibility. The random variable b_n^{in} represents the number of data packets after compression, i.e., $b_n^{in} = \Psi(d_n, k_n)$.

Similar to [22], the required energy to compress d data packets at each compression level k is calculated according to (1):

$$E_{CMP}(k, d) = d * E_{cmp}^{k} \tag{1}$$

where E_{cmp}^{k} is the amount of consumed energy to compress one data packet at compression level k in terms of Microjoules. To calculate E_{cmp}^{k}, we need the required energy to compress one bit of data at level k (denoted by e_{cmp}^{k}). Typical values for this energy is reported in [14]. We consider a linear generalization of the values in [14] to obtain the energy required to compress a data packet of L bits in size.

2.4. Transmission data queue model

The compressed data enters a queue of size B packets before transmission. The queue state information (QSI) is represented by a finite discrete state space $\mathcal{B} = \{0, 1, 2, \dots, B\}$, and the QSI dynamics can be expressed as (2):

$$b_{n+1} = min\left\{\max\left\{b_n - b_n^{out}, 0\right\} + b_n^{in}, B\right\} \tag{2}$$

where the number of packets currently in the buffer is shown as $b_n \in \mathcal{B}$, and the number of packets sent out from the buffer (during timeslot n) is denoted by b_n^{out}. Recall that b_n^{in} is determined by "compression control", while b_n^{out} is determined by "transmission control".

2.5. Wireless channel model

The buffered data must be transmitted over a wireless channel with potentially varying channel conditions. To model the time-varying quality of the channel, we use h_n to denote the quality state of the link connecting the node to the IoT gateway in the nth timeslot. In general, the evolution of fading channels can be modeled as a first-order finite state Markov chain (FSMC) (e.g., see [34]). Such a first-order model is known to be accurate for packet-level studies [35]. In this model, the SNR range is discretized into M distinct regions and then mapped into a finite-state space $\mathcal{H} = \{\hbar_1, \hbar_2, \dots, \hbar_M\}$. More precisely, suppose a set Γ of $M + 1$ SNR thresholds: $\Gamma = \{(-\infty, \Gamma_1), [\Gamma_1, \Gamma_2), \dots, [\Gamma_M, \infty)\}$. Assume $h_n = \alpha$, where α is the instantaneous SNR associated with the link. If α satisfies $\Gamma_m \leq \alpha < \Gamma_{m+1}$, the channel is said to be in state $\hbar_m$. Also, when a node probes the channel, the steady-state probability of being in the mth state is given by: $\upsilon_m = \int_m^{\Gamma_{m+1}} g(\alpha)\, d\alpha,\ m = 1, 2, \dots, M$, where, $g(\alpha)$ is the probability density function (PDF) of α.

Now, following the discussion in [36], the power required for a reliable and error-free transmission of b_n^{out} packets (each L bits in size) over a link with bandwidth W when the channel state is h_n is calculated by Eq. (3):

$$P\left(h_n, b_n^{out}\right) = \frac{W N_0}{h_n}\left(2^{\frac{b_n^{out} L}{W}} - 1\right) \tag{3}$$

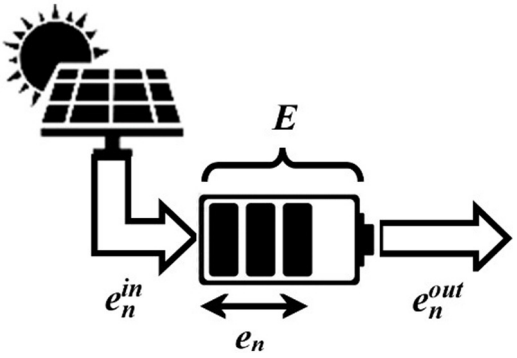

Fig. 2. Energy harvesting and storage model in IoT node.

where N_0 represents white Gaussian noise with mean zero and variance N_0^2, and the product of $W * N_0$ is normalized to 1. As can be seen from (3), in this model, the transmission power is a strictly *increasing* function of b_n^{out}. Eq. (3) is written assuming that h_n is measured in dB. After unit conversion, the power equation can be re-written as (4):

$$P\left(h_n, b_n^{out}\right) = \frac{W N_0}{10^{h_n/10}} \left(2^{\frac{b_n^{out} L}{W}} - 1\right) \tag{4}$$

Accordingly, the energy required to send a total of b_n^{out} data packets is calculated as a product of power and time according to (5), where τ denotes the time slot duration:

$$E_{TX}\left(h_n, b_n^{out}\right) = P\left(h_n, b_n^{out}\right) * \tau, \tag{5}$$

2.6. Energy source model

It is assumed that the IoT node is powered by solar energy [37] (See Fig. 2) according to a two-state process (active and inactive), which is a common assumption in the literature [38]. The two-state model is a reasonable approximation for describing solar harvesting where the harvester may be shaded/cloudy or clear. More specifically, in [37], the authors have empirically measured solar energy and fitted it to a stationary first-order Markovian model, in which the harvested solar energy is quantized into two states. The node collects energy at rate (power) $\not{2}$ in the active state and does not collect any energy in the inactive state. Following the model in [37], the time durations for which the source stays in active and inactive states can be assumed to be independent exponential distributions with parameters μ_a and μ_i, respectively.

Assumption 2 (*Energy Source Model*)**.** The random amount of harvested energy is denoted by $\{e_n^{in}\}_{n\in\mathbb{N}}$, and is modeled as the states of a two-state Markov chain, which take values from the finite space $\mathcal{X} = \{0, \not{2}\}$. ■

2.7. Energy storage model

The energy harvested from the environment is stored in an energy storage device (i.e. rechargeable battery or a super-capacitor), with capacity β. In principle, any inefficiency in the charge/discharge process can be factored into the energy consumption model. Another imperfection of the storage units could be their energy leakage. Here we assume that the storage unit is perfect in terms of leakage, as it is commonly assumed in the literature [39]. In most cases, this is a reasonable assumption, since the leakage is only a secondary effect.

Let e_n denote the actual amount of energy stored in the device at time n. In order to define the state space of the energy storage (battery state information or BSI), we uniformly quantize the energy buffer occupancy into $E = \beta/\Lambda$ levels denoted by $\mathcal{E} \stackrel{\text{def}}{=} \{1, 2, \dots \epsilon \dots, E\}$, where each level ϵ corresponds to the state where e_n is within the interval $[(\epsilon - 1)\Lambda, \epsilon\Lambda)$. The BSI dynamics can be expressed as

$$e_{n+1} = \frac{min(\max\left(e_n - e_n^{out}, 0\right) + e_n^{in}, \beta)}{\Lambda}$$

where e_n^{out} denotes the actual amount of energy consumed in timeslot n, which is determined by the "communication control decision" regarding the number of transmitted data packets as well as the selected compression level.

3. Joint compression and transmission control problem

Given the system model in Section 2, the goal is to design a joint compression and transmission control policy that minimizes the expected cumulative power expenditure of the device while satisfying a certain time average latency constraint for the sensory events queued in the transmission buffer. The control policy should be adaptive to the dynamic states of the system, i.e., to the volume of the sensory data (DSI), the wireless channel quality (CSI), the number of energy packets harvested from the environment (ESI), the energy level of the battery (BSI) as well as the occupancy state of the transmission data buffer (QSI). In particular, adaptation to CSI is needed to opportunistically exploit the channel dynamics and gain more value for the power invested. DSI and QSI-adaptability is needed to make the policy delay-aware under the conditions of unsaturated traffic and finite-length queue at the node. Finally, given that the node relies on energy harvesting for its operation, the control policy is subject to instantaneous energy availability constraint. An ESI and BSI-adaptive policy avoids inadvertent consumption of the harvested energy, and increases the odds that on urgent occasions, a larger amount of energy is available to power the node's compression/transmission tasks.

Also, under the stochastic dynamics of the system states, making myopic (i.e., instantaneously greedy) decisions about the compression level and transmission rate, based only on immediate costs, cannot lead to an optimal policy. For example, if the IoT node greedily minimizes its instantaneous cost, it tends to consume as little energy as possible to just barely satisfy the delay constraint in each timeslot. However, it may not take long before this naïve policy bumps into problems. For example, when the channel condition deteriorates or there is a sharp decrease in the amount of harvested energy, we regret not having consumed more energy in previous timeslots so that now, in our tight situation, we would be less concerned about violating the constraint on "average" transmission queue length. Our regret would even be higher when there is also a momentous increase in the intensity of sensory events leading to a large build-up in the transmission queue.

In fact, "communication control" is a sequential decision problem in the sense that it would be wise for the IoT node to occasionally make instantaneously suboptimal decisions in some timeslots, but by inducing the system states to transition to desirable states, it can obtain better long-term performance. Finding optimal policies in sequential decision problems can be done systematically by formally casting the problem in a stochastic optimization framework. Given the Markovian nature of our setting, we formulate the communication control problem for a delay-constrained IoT device as a Constrained Markov Decision Process (CMDP) [31]. Our objective is to minimize the long-term expected cumulative energy consumption of the IoT node subject to a long-term average constraint on the transmission queue length.

3.1. CMDP-based formulation

The CMDP associated with the "communication control" problem is defined as a tuple $\langle \mathcal{S}, A, \mathbb{P}, C_E, C_B \rangle$ where $S, A, \mathbb{P}, C_E$, and C_B denote respectively: the set of states, the set of control actions, transition probabilities, energy cost, and buffering cost. More specifically, the system state space is a discrete and finite space which is defined as the Cartesian product of CSI, DSI, QSI, ESI, and BSI spaces; i.e.,

$\mathcal{S} = \mathcal{H} \times \mathcal{D} \times \mathcal{B} \times \mathcal{E} \times \mathcal{X}$, and $s_n = (h_n, d_n, b_n, e_n^{in}, e_n) \in \mathcal{S}$ represents the system state at time n.

In the following, we elaborate on the control actions, the state transition law, the form of immediate cost function, the buffer immediate constraint, as well as the formulation of the stochastic optimization problem.

• **Actions:** In each timeslot, the decision-maker observes the current system state s_n and determines the set of feasible control actions in s_n. The space of possible actions in s_n is a discrete and finite space $A_{s_n} = \mathcal{K}_{s_n} \times \mathcal{B}_{s_n}$ in which $\mathcal{K}_{s_n} \subseteq \mathcal{K}$ is the set of feasible compression levels (which depends on DSI d_n and BSI e_n) and $\mathcal{B}_{s_n} \subseteq \mathcal{B}$ is the set of data packets that can be transmitted out of the buffer (which depends on the CSI h_n, BSI e_n as well as QSI b_n). More formally, using an indicator function notation, we may determine whether a joint control action (k, b^{out}) can be included in the feasible sets $\mathcal{K}_{s_n}$ and $\mathcal{B}_{s_n}$:

$$\mathbb{I}\left(k \in \mathcal{K}_{s_n}, b^{out} \in \mathcal{B}_{s_n}\right) = \begin{cases} 1 & if \quad E_{CMP}\left(k, d_n\right) + E_{TX}\left(h_n, b^{out}\right) \le e_n * EU \text{ and } b^{out} \le b_n \\ 0 & otherwise \end{cases} \tag{6}$$

According to Eq. (6), the sum of the required amount of compression and transmission energy is compared with the available energy level in the battery to identify the feasible actions. Each node can only take the actions for which there is enough energy. The current action of the decision-maker is denoted as a_n which is defined by the tuple $a_n = (k_n, b_n^{out}) \in A_{s_n}$.

• **System transition law:** Following the execution of the action a_n in state s_n, the system state transits probabilistically to the next state s_{n+1}. The transition probability $\mathbb{P}\left(s_{n+1}|s_n, a_n\right)$ is determined by Eq. (7):

$$\mathbb{P}\left(s_{n+1}|s_n, a_n\right) = \mathbb{P}\left(h_{n+1}|h_n\right).\mathbb{P}\left(b_{n+1}|b_n, d_n, b_n^{out}\right).\mathbb{P}\left(d_{n+1}|d_n\right).\mathbb{P}\left(e_{n+1}|e_n, e_n^{in}, a_n\right). \mathbb{P}\left(e_{n+1}^{in}|e_n^{in}\right). \tag{7}$$

Note that in the above transition law, the impact of CSI h_n on the next QSI b_{n+1} as well as on the next BSI e_{n+1} is implicit in its role in determining the feasible action $a_n = (k_n, b_n^{out})$.

• **Immediate cost function:** The immediate cost of taking action a_n in state s_n is defined to be the total energy consumption cost denoted by $C_E\left(s_n, a_n\right)$ and calculated by Eq. (8):

$$C_E\left(s_n, a_{s_n}\right) = E_{CMP}\left(k_n, d_n\right) + E_{TX}\left(h_n, b_n^{out}\right). \tag{8}$$

• **Immediate constraint function:** We define the immediate constraint function $C_B\left(s_n, a_n, s_{n+1}\right)$ in a way that can keep the system from violating a certain reporting delay threshold for the sensory events. To this end, we apply the Little's law [36] to obtain a measure of the mean latency experienced by the reported events. According to Little's law [36], the average buffer length, $\bar{C}_B$, is equal to the product of the average arrival rate of sensory data, $\bar{a}$, and the average latency experienced by the packets in the buffer, $\bar{D}$, as shown in Eq. (9):

$$\bar{C}_B = \bar{a}\bar{D}. \tag{9}$$

Hence, by ignoring the constant $\bar{a}$, we can use $\bar{C}_B$ itself as a good measure of the average delay $\bar{D}$. Now, since the mean packet delay in the buffer should not exceed a certain threshold, we need to keep track of the instantaneous buffer occupancy as an immediate constraint (10):

$$C_B\left(s_n, a_n, s_{n+1}\right) \triangleq b_{n+1}. \tag{10}$$

• **The optimization objective:** The objective in our CMDP formulation is to minimize the long-term average discounted energy consumption, $\bar{C}_E$, while maintaining the average discounted buffer length $\bar{C}_B$ under a given application-specific threshold δ. The threshold δ is assumed to be specified by the system designer to reflect the amount of delay tolerance in practical applications.

More formally, the IoT node seeks to learn an optimal policy π^* to control its compression level and transmission rate, so that for all $s \in S$ we have:

$$\begin{aligned} &\pi^*(s) \in \underset{\pi}{argmin}\ \bar{C}_E^{\pi}(s) \triangleq \mathbb{E}^{\pi}\left[\sum_{n=1}^{\infty} \gamma^n C_E\left(s_n, a_{s_n}\right) | s_1 = s\right] \\ &s.t.\ \bar{C}_B^{\pi} \triangleq \mathbb{E}^{\pi}\left[\sum_{n=1}^{\infty} \gamma^n C_B\left(s_n, a_{s_n}, s_{n+1}\right) | s_1 = s\right] \le \frac{\delta}{1-\gamma} \end{aligned} \tag{11}$$

where the functions $\bar{C}_E^{\pi}$ and $\bar{C}_B^{\pi}$ are average discounted sums, in which the discount factor $0 \le \gamma < 1$, represents the importance of the future costs in proportion to the current costs. Setting smaller values for γ gives less importance to the future costs, thus rendering the controller's behavior towards a more myopic policy. Of particular note here is the modification done on the tolerable delay threshold δ in the right hand side of the constraint term in (11). In fact, since we assume that in most applications, the system delay is typically dictated in terms of a "time average" not a "discounted sum", the application-specified threshold δ is converted here to its equivalent cumulative discounted value via dividingδ by the infinite geometric series with growth rate γ. This way, while our overall formulation in (11) is kept in compliance with a standard discounted formalism, satisfying this modified constraint would now be equivalent to satisfying the delay threshold in the "time average" sense.

3.2. The Lagrangian technique

We apply the standard Lagrangian technique [40] to convert the problem (11) to its unconstrained counterpart. A similar scheme has also been presented in [41] in another context. More specifically, we introduce a Lagrange multiplier $\lambda \ge 0$ to linearly combine the objective function in (11) with its constraint, and define a new combined cost function called "Lagrangian" $\bar{\mathcal{L}}^{\lambda,\pi}(s)$ which is expressed as (12):

$$\bar{\mathcal{L}}^{\lambda,\pi}(s) \triangleq \bar{C}_E{}^{\pi}(s) + \lambda\left(\bar{C}_B^{\pi} - \delta\right) \quad \forall s \in \mathcal{S} \tag{12}$$

Intuitively, in (12), if the average discounted buffer-length $\bar{C}_B^{\pi}$ exceeds the threshold value δ, the resultant difference would be added by a positive coefficient to the system cost to further penalize the controller.

Given the nature of $\bar{C}_E{}^{\pi}$ and $\bar{C}_B^{\pi}$, the function $\bar{\mathcal{L}}^{\lambda,\pi}(s)$ is also a long-term average and the immediate Lagrangian $l(s, a, \lambda)$ corresponding to it can be defined as (13):

$$l(s, a, \lambda) = C_E(s, a) + \lambda(C_B\left(s_n, a_n, s_{n+1}\right) - \delta) \tag{13}$$

Now, according to (13), Lagrangian in Eq. (12) can be rewritten as (14):

$$\bar{\mathcal{L}}^{\lambda,\pi}(s) \triangleq E^{\pi}\left[\sum_{n=1}^{\infty} \gamma^n l\left(s_n, a_n, \lambda\right) | s_1 = s\right] \forall s \in \mathcal{S} \tag{14}$$

From (12), we note that for a feasible policy π, we would have:

$$\bar{C}_B^{\pi}(s) - \delta \le 0\,(\forall s \in \mathcal{S}). \tag{15}$$

As a result,

$$\bar{\mathcal{L}}^{\lambda,\pi}(s) \le \bar{C}_E{}^{\pi}(s) \quad \forall s \in \mathcal{S},\ for\ all\ feasible\ \pi. \tag{16}$$

Hence, by minimizing the above inequality over the space of all feasible policies, we have:

$$\min_{\pi} \bar{\mathcal{L}}^{\lambda,\pi}(s) \le \min_{\pi} \bar{C}_E{}^{\pi}(s) \quad \forall s \in \mathcal{S},\ for\ all\ feasible\ \pi. \tag{17}$$

Therefore, $min_{\pi}\ \bar{\mathcal{L}}^{\lambda,\pi}(s)$ gives a lower bound for the value of the objective function in the constrained problem given in (11). By finding the greatest lower bound, i.e., by maximizing λ in $min_{\pi}\ \bar{\mathcal{L}}^{\lambda,\pi}(s)$, the best lower bound can be obtained for the constrained problem (11). Therefore, instead of solving (11), we may now tackle with the following new unconstrained optimization problem ([31], Theorem 3.6):

$$\max_{\lambda} \min_{\pi} \bar{\mathcal{L}}^{\lambda,\pi}(s), \quad \forall s \in \mathcal{S} \tag{18}$$

Define the pair (π^*, λ^*) as a solution to (18). According to [26], if the immediate cost $C_E(s_n, a_n)$ and the immediate constraint $C_B(s_n, a_n, s_{n+1})$ are both convex, there is no difference between the value $\bar{C}_E^{\pi}$, obtained from the constrained problem (11), and $\bar{\mathcal{L}}^{\lambda^*,\pi^*}$, obtained from (18). In our formulation, both functions $C_E(s_n, a_n)$ and $C_B(s_n, a_n, s_{n+1})$ are indeed convex. Consequently, by solving the two optimization problems given in (19) and (20), the optimal policy can be achieved:

$$\pi^* \in arg \min_{\pi} \bar{\mathcal{L}}^{\lambda,\pi} \tag{19}$$

$$\lambda^* \in arg \max_{\lambda} \bar{\mathcal{L}}^{\lambda,\pi^*} \tag{20}$$

In Section 4, we propose a model-free reinforcement learning technique to calculate the optimal solution pair (π^*, λ^*).

4. Learning the optimal control policy

In practical scenarios, it is often difficult or even impossible to obtain reliable statistical information about the underlying stochastic processes governing the system dynamics; e.g. the average intensity of the event flow, energy harvesting, channel quality variations, etc. In these situations, we cannot adopt a model-based scheme (e.g., standard value iteration [42] or policy iteration [43]) for solving the problem in (18), since such a scheme relies on the knowledge of the system's transition kernel in (7).

Alternatively, in this section, we propose a model-free learning algorithm to compute the optimal solution pair (π^*, λ^*) in the absence of the statistical knowledge of the system, and instead, by relying only on the immediate feedbacks (in the form of instantaneous values of cost, constraint, and state observation sequence) acquired through real-time interactions with the operating environment.

More specifically, we equip the IoT node with a learning algorithm consisting of a coupled recursion that iteratively estimates π^* and λ^* for simultaneously solving (19) and (20).

In particular, the problem (19) is solved iteratively using Q-learning [25,44] to gradually estimate the optimal policy π, while treating λ as effectively quasi-static. For completeness, however, we begin with presenting Bellman equations [45] associated with our problem, and then work our ways towards Q-learning. Following standard definitions, we denote the so-called Q function by $Q^{*,\lambda}(s, a)$, which is defined as the sum of the immediate Lagrangian $l(s, a, \lambda)$, obtained by taking action a in a current state s plus the long-term Lagrangian obtained from following the (unknown) optimal policy π^* thenceforward; i.e.,

$$Q^{*,\lambda}(s, a) = l(s, a, \lambda) + \gamma \sum_{\acute{s} \in \mathbb{S}} \mathbb{P}(\acute{s}|s, a)\, \bar{\mathcal{L}}^{\lambda,\pi^*}(\acute{s}) \tag{21}$$

where,

$$\bar{\mathcal{L}}^{\lambda,\pi^*}(s) = \min_{a \in \mathcal{A}(\acute{s})} Q^{*,\lambda}(s, a), \quad \forall s \in \mathbb{S} \tag{22}$$

The optimal policy $\pi^{*,\lambda}(s)$ (parameterized with λ) is then obtained as:

$$\pi^{*,\lambda}(s) = arg \min_{a \in \mathcal{A}(s)} Q^{*,\lambda}(s, a), \quad \forall s \in \mathbb{S} \tag{23}$$

While a model-based scheme needs the knowledge of $\mathbb{P}$ for solving the Bellman equation in (21), Q-learning relieves us from such necessity through repeated estimation of $Q^{*,\lambda}(s, a)$ for all pairs (s, a). In the beginning, the IoT node initializes an estimate table $\hat{Q}_n(s, a)$ for all (s, a) pairs with zero or arbitrary values. In each iteration, the node observes the current system state s and chooses an action a from its action space. The node implements the selected action, computes its immediate reward as $l(s, a, \lambda)$, and then updates its estimate $\hat{Q}_n(s, a)$ for the current pair (s, a) according to the following rule:

$$\hat{Q}_n(s, a) \longleftarrow (1 - \beta_n)\, \hat{Q}_{n-1}(s, a) + \beta_n \left[l(s, a, \lambda) + \gamma \min_{\acute{a} \in \mathcal{A}(\acute{s})} \hat{Q}_{n-1}(\acute{s}, \acute{a}) \right] \tag{24}$$

In (24), $\hat{Q}_n(s, a)$ represents the time nth estimate of $Q^{*,\lambda}(s, a)$ and β_n denotes the step size (learning rate) of Q-learning, which is computed for each pair (s, a) according to (25):

$$\beta_n = \frac{1}{1 + (visit_n(s, a))^{0.65}} \tag{25}$$

where $visit_n(s, a)$ is the number of times the pair (s, a) has been observed up to iteration n.

The only necessary condition to guarantee the convergence of the above Q-learning algorithm is to choose actions at each step in a so-called ε-greedy fashion (e.g., see [25]). The symbol ε denotes a small probability with which a reinforcement learning agent explores the unknown environment from time to time. In particular, while the controller needs to generally take greedy actions according to the optimal policy being estimated (i.e., $\pi^{*,\lambda}(s)$ from (23)), it occasionally needs to take random (yet feasible) actions to further explore the quality of alternative choices, so as not to get biased towards actions with "deceptively" good Q values. Using such ε-greedy action selection policy, the described Q-learning algorithm is guaranteed to converge to $\pi^{*,\lambda}(s)$ and along with it, to the minimum Lagrangian $\bar{\mathcal{L}}^{\lambda,\pi^*}$ for a constant value λ (c.f., Theorem 1).

Now, we turn attention towards computing the optimal Lagrange multiplier. Knowing $\bar{\mathcal{L}}^{\lambda,\pi^*}$, We note that (20) is a maximization problem, and had we known the knowledge of the transition laws, (20) could have been solved by setting to zero the derivative of (12) w.r.t. λ. In the absence of $\mathbb{P}$, however, we instead need to deploy an iterative scheme to estimate λ^*. In particular, we define $\hat{\lambda}_n$ as the nth estimate of λ^*, and use stochastic sub-gradient ascent algorithm for directing $\hat{\lambda}_n$ towards its optimal value; i.e.,

$$\hat{\lambda}_{n+1} = \Omega\left[\hat{\lambda}_n + \alpha_n (C_B(s_n, a_n, s_{n+1}) - \delta)\right] \tag{26}$$

where α_n serves as the step size (learning rate) which is computed by Eq. (27):

$$\alpha_n = \frac{1}{n + 1} \tag{27}$$

The operator $\Omega[.] \overset{\text{def}}{=} \max(., 0)$ is a projection operator and is meant for keeping $\hat{\lambda}_n$ from ever becoming negative. The term $(C_B(s_n, a_n, s_{n+1}) - \delta)$ is an instantaneous estimate of the gradient direction for the function $\bar{\mathcal{L}}^{\lambda,\pi^*}$. Its form corresponds to the derivative of (12) w.r.t. λ, but unlike an exact derivative, it is only a noisy instantaneous estimate. This justifies our application of "stochastic" sub-gradient method to guarantee gradual convergence towards optimal λ^*.

One subtlety remains that needs further clarification. It concerns the concurrent estimation of λ^* and Q^* using the learning rules (24) and (26). It should be noted that $\hat{\lambda}_n$ and $\hat{Q}_n$ are coupled together by definition. However, we have allowed for their simultaneous updating, which seemingly work against each other by creating a moving target for one another. The elegant way to proceed these two updates concurrently, while yet avoiding divergence, is to operate these two recursions on two different time scales. More formally, we choose the update rates for these two estimates, i.e. $\beta(n)$ and $\alpha(n)$, such that they satisfy some standard conditions from the theory of two-timescale stochastic approximation (e.g., see [33]):

$$\sum_n \left(\beta(n)^2 + \alpha(n)^2\right) < \infty, \quad \lim_{n \to \infty} \frac{\alpha(n)}{\beta(n)} \to 0 \tag{28}$$

If the conditions in (28) are satisfied, the estimate $\hat{\lambda}_n$ will be updated at a slower time-scale compared to $\hat{Q}_n$. This way, $\hat{Q}_n$ appears to be equilibrated (or convergent) in the eyes of the update rule for $\hat{\lambda}_n$. While, from the viewpoint of $\hat{Q}_n$, the estimate $\hat{\lambda}_n$ appears to be quasi-static [30].

Armed with these understandings, the pseudo code of the learning algorithm to find the optimal control policy for joint compression level and transmission rate is presented in Table 1.

Rigorous proofs in the context of generic CMDP formulations concerning the convergence of the estimated pair $(\hat{Q}_n, \hat{\lambda}_n)$ to their optimal

Table 1
Pseudo code of the learning algorithm.

// Initialization
$n = 0$ (initial time index)
$\hat{Q}_n(s,a) = 0 \quad \forall s, a$
$\hat{\lambda}_n = 0$
$counter = 0$ (state-action counter)
γ = some discount factor
δ = some (time) average buffer threshold
initialize $s_0 = (h_0, d_0, b_0, e_0^{in}, e_0)$ arbitrarily
Identify feasible actions $A(s_1)$ using (7).
// Main Learning Loop
repeat
1. Select an action a_{s_n} from the set $A(s_n)$ according to an ε -greedy policy $\hat{\pi}_n$:
   $a_{s_n}^{greedy} = \arg\min_a \hat{Q}_n(s_n, a)$ // the greedy action according to current Q-estimate
   $i = index\left(a_{s_n}^{greedy}, A(s_n)\right)$ // i denotes the index of the greedy action in the set $A(s_n)$
   $\hat{\pi}_n(s_n) = (1-\varepsilon).\boldsymbol{e}_i + \frac{\varepsilon}{|A(s_n)|}.\mathbf{1}_{|A(s_n)|}$ // $\boldsymbol{e}_i$ is the i-th basis vector in $\mathbb{R}^{|A(s_n)|}$ and $\mathbf{1}_{|A(s_n)|}$ is a $|A(s_n)| \times 1$ vector of 1's.
   $a_{s_n} = (k_n, b_n^{out}) \sim \hat{\pi}_n(s_n)$ // the action a_{s_n} is sampled from $\hat{\pi}_n(s_n)$
2. Update $counter$.
3. Compute $C_E(s_n, a_n)$ cost using (9).
4. Compute $C_B(s_n, a_n)$ cost using (11).
5. Compute Lagrangian $l(s,a,\lambda)$ using (14).
6. Observe the next system state s_{n+1}.
7. Identify feasible actions $A(s_{n+1})$ using (7).
8. Compute step size β_n using (26).
9. Update $\hat{Q}_n(s,a)$ using (25).
10. Compute step size α_n using (28).
11. Update $\hat{\lambda}_{n+1}$ using (27).
12. Update n.

until $n < MAX\_iteration\_number$

values (Q^*, λ^*) can be found in the literature on stochastic approximation (e.g., see [33,46]). Here, we refrain from delving into such details for the sake of brevity. Instead, we give an outline of the convergence proof in the form of Theorem 1 below:

Theorem 1. *With the update rates $\alpha(n)$ and $\beta(n)$ chosen as (28), as $n \to \infty$, $\hat{\lambda}_n \to \lambda^*$, $\hat{Q}_n \to Q^*$, and $\hat{\pi}_n \to \pi^*$.*

Proof (*Outline*). According to ([31], Theorem 3.6), solving the constrained problem given in (11) is equivalent to solving the unconstrained max–min problem given in (18). The min in this max–min problem is in fact achieved by some policy which is optimal for (11). Under the mild Slater condition for the feasibility of convex minimization problems [26], there exists an optimal Lagrange multiplier λ^* such that the optimal solution of the CMDP $\langle \mathcal{S}, A, \mathbb{P}, C_E, C_B \rangle$ is equivalent to the optimal solution of the unconstrained MDP $\langle \mathcal{S}, A, \mathbb{P}, l(s,a,\lambda^*) \rangle$. Evidently, for any "fixed" $\lambda \geq 0$, the MDP $\langle \mathcal{S}, A, \mathbb{P}, l(s,a,\lambda) \rangle$ can be solved using Q-learning with an ε-greedy exploration policy [32] to achieve convergence of the form: $\hat{Q}_n \overset{n\uparrow\infty}{\to} Q^{*,\lambda}$, and $\hat{\pi}_n \overset{n\uparrow\infty}{\to} \pi^{*,\lambda}$. Hence, the Lagrangian $\bar{\mathcal{L}}^{\lambda,\hat{\pi}_n}(s), \forall s \in \mathcal{S}$ would also converge to $\bar{\mathcal{L}}^{\lambda,\pi^*}(s), \forall s \in \mathcal{S}$ (c.f., Eq. (19)). Now, using the two-timescale analysis in [33] and in view of $\beta(n) = o(\alpha(n))$, in the coupled recursions of and Q-learning for pushing the sequence $\{\hat{Q}_n\}_{n\in\mathbb{N}}$ towards Q^* and stochastic sub-gradient for driving $\{\hat{\lambda}_n\}_{n\in\mathbb{N}}$ towards λ^*, the sequence of multipliers $\{\hat{\lambda}_n\}_{n\in\mathbb{N}}$ is "effectively freezed" at someλ. This way, the convergence of Q-learning towards its limit would not be compromised by the time-varying sequence $\{\hat{\lambda}_n\}_{n\in\mathbb{N}}$. What remains to be shown is the convergence of $\{\hat{\lambda}_n\}_{n\in\mathbb{N}}$ to $\lambda^* \in arg\ max_\lambda\ \bar{\mathcal{L}}^{\lambda,\pi^*}$. Note that for $\forall s \in \mathcal{S}$, $\bar{\mathcal{L}}^{\lambda,\pi^*}(s)$ is a function of λ only. Similarly to [46], it can be argued that mapping $\lambda \to \bar{\mathcal{L}}^{\lambda,\pi^*}(s)$ is piecewise linear and concave for $\forall s \in \mathcal{S}$. Let ∇_λ denote the gradient in the λ variable. In order to characterize the limiting behavior of $\{\hat{\lambda}_n\}_{n\in\mathbb{N}}$, we may use an argument based on the well-established ordinary differential equation (ODE) approach [33] which treats stochastic approximation (26) as a noisy discretization of an autonomous ODE of the form (29):

$$\frac{d\lambda(t)}{dt} = \nabla_\lambda \bar{\mathcal{L}}^{\lambda,\pi^*}(s), \forall s \in \mathcal{S}, \tag{29}$$

It then holds that the stochastic (sub-)gradient iterations on $\{\hat{\lambda}_n\}_{n\in\mathbb{N}}$ track the differential equation in (29), and therefore converges to the set of maxima of $\bar{\mathcal{L}}^{\lambda,\pi^*}$ [46]. Combining this with the abovementioned discussion yields that $(\hat{Q}_n, \hat{\pi}_n, \hat{\lambda}_n)$ converges to (Q^*, π^*, λ^*), which is an alternative way of saying $\bar{\mathcal{L}}^{\lambda,\hat{\pi}_n}(s)$ converges to $\bar{\mathcal{L}}^{\lambda^*,\pi^*}(s)$ for all $s \in \mathcal{S}$. ■

5. Performance evaluation and results

In this section, we implement our proposed algorithm in a simulation environment where we simulate the point-to-point scenario depicted in Fig. 1. We first explain the simulation setup including the experiment settings and simulation parameters in Section 5.1. Next, in Section 5.2, as the first set of experiments, we investigate the convergence properties of the learning algorithm. In Section 5.3, we present the simulation results comparing the performance of the proposed algorithm against some baseline schemes. We also evaluate how the variations in system parameters influence the performance. In particular, we show how the average energy consumption varies under different regimes of sensory data arrival rate. Also, we investigate the impact of energy charging rate on average buffer length, as well as the impact of energy buffer capacity on the average energy consumption. Finally, in Section 5.4 we explore the impact of the quantization accuracy for the energy storage space on the system's performance.

5.1. Simulation parameters

We simulate a time-slotted system with slot duration of $\tau = 1$ ms. Although our algorithm does not depend on any distribution for the channel SNR α, for the purpose of modeling, we simulate slow Rayleigh channels for each link. For a Rayleigh model, channel SNR α is

Table 2
Simulation parameters.

Parameter	Value	Description
τ	1 (ms)	Timeslot duration
n	2e6	Number of iterations
$\mathcal{P}$	2 (mW)	Harvesting power [42]
β	3 (mWh)	Energy storage capacity (3F NESSCAP supercapacitor) [47]
E	12	Default energy storage quantization levels
μ_a	1/43 200 Hz	"Active" (harvesting) duration
μ_i	1/43 200 Hz	"Inactive" (harvesting) duration
D	5 (pkts)	Maximum number of data packets sensed in each timeslot
L	100 (bytes)	Size of data packets (A typical IEEE802.15.4 packet contains data up to 132 bytes [48])
B	9 (pkts)	Data buffer capacity
K	6	Maximum compression level
e^k_{cmp}	[0,0.360,0.380,0.440,0.540,0.600] (nJ)	Energy consumption to compress a bit at level k [23]
$\mathcal{H}$	[−13,−8.47,−5.41,−3.28,−1.59,−0.08,1.42,3.18] (dB)	Wireless channel state space [34]
W	5 MHz	Channel bandwidth
γ	0.85	Discount factor
δ	4.15 pkts	Buffer threshold
ε	0.1	The probability of random selection in ε-greedy method
$\hat{Q}_0(s,a)$	0	The initial value of Q
$\hat{\lambda}_0$	0	The initial value of Lagrange multiplier

an exponentially distributed random variable with probability density function given by $g(\alpha) = \frac{1}{\bar{\alpha}} e^{\frac{-\alpha}{\bar{\alpha}}}$, where $\bar{\alpha} = \mathbb{E}[\alpha]$ is the average SNR. As discussed in Section 2.5, the proposed algorithm is not aware of the perfect instantaneous CSI α; however, we assume that only finite CSI is fed back, and the node only knows α belongs to an interval $\left[\Gamma_m, \Gamma_{m+1}\right)$ instead of having the exact value. Hence, similarly to [34], we discretize the channel into eight equal probability bins, with the boundaries specified by:

$$\left\{\begin{matrix}(-\infty, -8.47\ \text{dB}), [-8.47\ \text{dB}, -5.41\ \text{dB}), & [-5.41\ \text{dB}, -3.28\ \text{dB}), [-3.28\ \text{dB}, -1.59\ \text{dB}), \\ [-1.59\ \text{dB}, -0.08\ \text{dB}), & [-0.08\ \text{dB},\ 1.42\ \text{dB}), [1.42\ \text{dB},\ 3.18\ \text{dB}), [3.18\ \text{dB},\ \infty)\end{matrix}\right\},$$

and select the channel space as follows: $\mathcal{H} = \{h_1 = -13\ \text{dB},\ h_2 = -8.47\ \text{dB},\ h_3 = -5.41\ \text{dB},\ h_4 = -3.28\ \text{dB},\ h_5 = -1.59\ \text{dB},\ h_6 = -0.08\ \text{dB},\ h_7 = 1.42\ \text{dB},\ h_8 = 3.18\ \text{dB}\}$. The fixed quantized average SNR value $\bar{\alpha}_m$ for each state $\mathcal{H}_m, m = 1, 2, \ldots, M$ then becomes $\bar{\alpha}_m = \left(\upsilon_m\right)^{-1} \int_{\Gamma_m}^{\Gamma_{m+1}} \alpha g(\alpha)\, d\alpha$, where following our discussion in Section 2.5, $\upsilon_m = e^{\frac{-\Gamma_m}{\bar{\alpha}}} - e^{\frac{-\Gamma_{m+1}}{\bar{\alpha}}}$. Similarly to [49,50], the transition probability matrix $\left(\mathbb{P}_{m,\acute{m}}\right)_{m,\acute{m}=1,\ldots,8}$ of the FSMC is assumed to have the following structure:

$$\mathbb{P} = \begin{bmatrix} \theta & \sigma & 0 & 0 & 0 & \cdots & 0 & \sigma \\ \sigma & \theta & \sigma & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma & \theta & \sigma & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma & 0 & 0 & 0 & 0 & \cdots & \sigma & \theta \end{bmatrix},$$

where $\theta = 1 - 2\sigma$ and $\sigma = \mathcal{O}(f_d \tau)$, with f_d and τ denoting the Doppler frequency shift and packet duration time, respectively. In FSMC model for slow fading, we assume that α evolves slowly with time; i.e., at time $n+1$, it is highly likely that α stays within the same region as it was at time n, and there is a negligible chance that it transitions to other regions. The product $f_d \tau$ characterizes the fading speed of the channel relative to the packet length. A small $f_d \tau$ means that the channel fading rate is small. Throughout simulations, different instances of the matrix $\mathbb{P}$ are used to generate the channel data profile.

For our simulation parameters to be chosen in a realistic fashion, we envisage a MICAz wireless sensor [51] energized by solar power [37]. In [37], the experimentally measured solar energy has been fitted to a stationary 1st-order Markov chain. In particular, the harvested solar energy is quantized into two states by setting 1.4 mW as the quantization threshold. Accordingly, we consider an active harvesting power be $\mathcal{P} = 2$ mW and the inactive harvesting power as 0mW in our experiments. We also simulate the scenario where $\mu_a = \mu_i = \frac{1}{43200}$ Hz. This means that the active-to-inactive or inactive-to-active transition occurs (on average) once within 12 h, respectively. This scenario is meant to mimic a sunny day when (on average) the system stays in the active and the inactive states for around 12 h each. As with the battery capacity, we assumed that a super-capacitor is used as the energy storage unit. Compared to batteries, super-capacitors have higher "power density" and lower "energy density"; hence, they can deliver the energy to the load more quickly. Moreover, super-capacitors are able to be charged very fast (a superiority over the batteries). However, as super-capacitors are in general more bulky compared to the batteries, we choose a tiny 2.7 V 3F Cell NESSCAP super-capacitor with weight 1.5 g [47], whose energy storage capacity is 3 mWh.

The simulation parameters are summarized in Table 2.

5.2. Convergence properties

Before discussing the convergence results, we first need to discern between two different notions of time used in the horizontal axes of the convergence plots. The first notion is called "episode" which does not correspond to the actual system time slots. In fact, since our problem is an instance of an infinite-horizon MDP, the so-called "value function" is state-dependent, and what we are interested in is the evolution of the estimates for the long-term average discounted value of a specific initial state s_0. Now, each "episode" actually marks every time slot at which the initial state s_0 has been revisited over the course of the execution of the algorithm. As such, it is more than likely that several time slots lie in between every two consecutive episodes. In fact, each time we bump into s_0, the sum of discounted rewards since the previous visit can be considered as one new sample of the value function at s_0. The moving average of these samples over the course of a number of episodes will converge to some steady value. The second notion of time which we call "iteration" corresponds to the actual system time slot. In what follows, the convergence plots which demonstrate the performance measures (energy consumption or buffer occupancy) are in terms of the number of episodes; in contrast, the algorithmic quantities (Q function as well as λ) have been plotted in terms of iterations.

The proposed learning algorithm solves a constrained optimization problem. Hence, to assess its convergence, we first evaluate the convergence of Lagrange multiplier as the main indicator of the algorithm convergence. The estimates of $\hat{\lambda}_n$, obtained from iterative updates in Eq. (26), are plotted in Fig. 3. As shown in this figure, the constraint Lagrange multiplier converges after approximately 40 000 iterations. Convergence to a non-zero value represents a case where the constraint on the average buffer length is satisfied in a tight manner.

Next, in Fig. 4, we plot the evolution of the Q values for four different state–action pairs. While an outline of the theoretical convergence is given in Theorem 1, our simulations demonstrate that convergence of the algorithm occurs in a reasonable number of iterations (time slots) for practical purposes. In fact, the slot duration in wireless systems is of the order of milliseconds, and in data transfer applications where

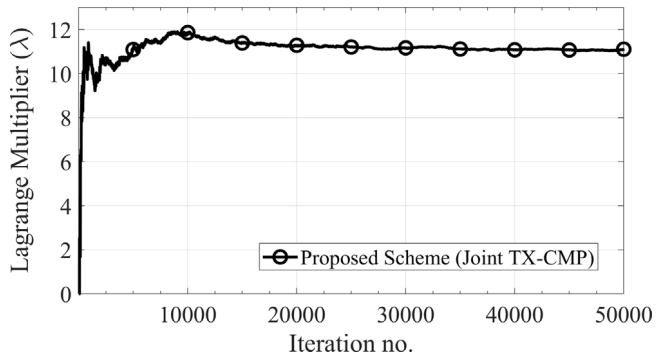

Fig. 3. Convergence of the Lagrange multiplier.

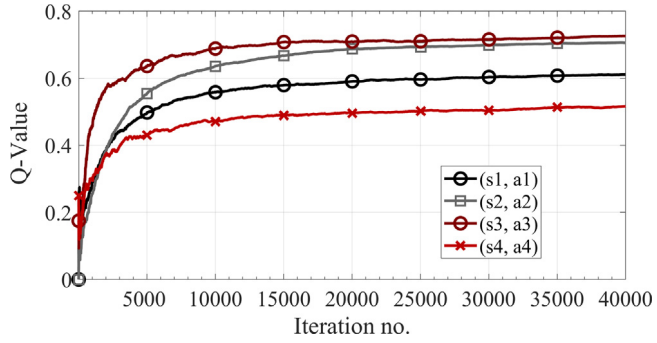

Fig. 4. Evolution of the Q-values.

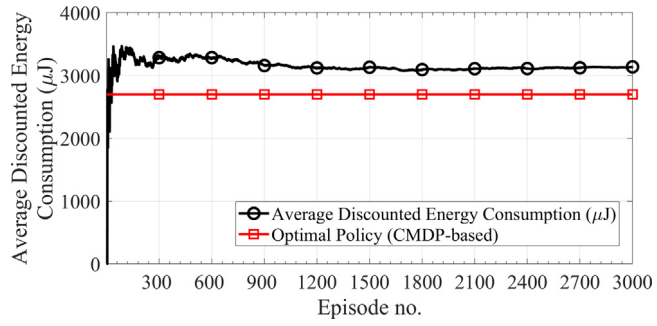

Fig. 5. The convergence of the average discounted energy consumption.

the duration of the transfer is of the order of tens of seconds, the sub-optimality of our algorithm may be there only for a fraction of data transfer.

Fig. 5 shows the convergence of the average discounted energy consumption $\bar{C}_E{}^{\pi}$ measured in micro-joules. From the figure, we notice that the algorithm converges to the cumulative value of 3076 μJ after 300 learning episodes. For the sake of comparison with a theoretical benchmark, we have utilized the technique of "occupation measures" from the CMDP literature (see [31]), to optimally solve problem (11). More details are given in Section 5.3. The optimal CMDP-based policy has been computed using the perfect model of the system statistics (i.e., energy charging process, event generation as well the channel state transition matrix). As can be seen in Fig. 5, the proposed model-free approach for joint optimization of compression and transmission can approximately reach to a 16% margin of its theoretical benchmark.

In the next experiment, we plot the evolution of the average buffer length across timeslots. As shown in Fig. 6, it is evident that under the control of our learning algorithm, the average buffer length does not

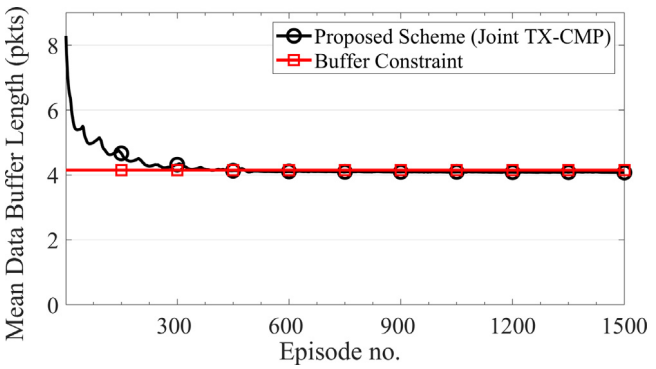

Fig. 6. Average buffer length satisfies its constraint.

violate the pre-specified threshold (red line), completely satisfying the constraint. In the experiment, the threshold value is set to 4.15 and the average buffer length reaches 4.09.

5.3. Comparison with baseline schemes

The proposed algorithm for joint compression and transmission control is provably convergent to an optimal solution (c.f., the discussion in Section 4). Hence, it is expected that it outperforms any competitive heuristic (suboptimal) algorithm for the same problem. In this section, we compare the performance of the obtained control policy against three baseline schemes. The considered policies are as follows:

- **Optimal policy (CMDP-based):** It has been shown in [31] that a using a technique called "occupation measures", a CMDP model can be solved using linear programming (LP) in polynomial time. Here, we briefly hint at the technique, but the interest reader may refer to [31] for detailed derivation. Occupation measure is defined as a probability measure over the set of state–action pairs and denoted by $\rho(s,a)$. The objective function and the constraints given in Eq. (11) can be expressed with respect to $\rho(s,a)$ as:

$$\rho^* = arg \min_{\rho} \sum_{s\in\mathcal{S}} \sum_{a\in A(s)} \rho(s,a) C_E(s,a) \tag{30}$$

$$s.t. \quad \sum_{s\in\mathcal{S}} \sum_{a\in A(s)} \rho(s,a) C_B(s,a) \le \delta \tag{31}$$

$$\rho(s,a) \ge 0, \quad \forall s \in \mathcal{S}, \forall a \in A(s) \tag{32}$$

$$\sum_{z\in\mathcal{S}} \sum_{a\in A(z)} \rho(z,a) \left(\mathbb{I}_s(z) - \gamma \mathbb{P}(s|z,a)\right) = (1-\gamma)\psi(s), \quad \forall s \in \mathcal{S} \tag{33}$$

  The function $\mathbb{I}_s(.)$ in (33) is the well-known Kronecker delta function where $\mathbb{I}_s(z) = 1$ if $s = z$ and $\mathbb{I}_s(z) = 1$ if $s \neq z$. Also, $\psi(.)$ is an initial distribution over the set of states (which we consider to be uniform for the sake of our experiments).
  The optimal stationary randomized policy $\pi^*(.|s)$ is then obtained from the optimal ρ^* according to the following equation:

$$\pi^*(a|s) = \frac{\rho^*(s,a)}{\sum_{a\in A(s)} \rho^*(s,a)}, \forall s \in \mathcal{S}, \forall a \in A(s)$$

  In sum, if the probabilistic model of the system (i.e., $\mathbb{P}(s|z,a)$ and $\psi(.)$) is perfectly known, it then follows from ([31], Theorem 3.2) that the optimal value of $\bar{C}_E^{\pi}$ can obtained using the above constrained program.
- **Optimal Compression (OPT-CMP) policy:** It only controls the compression level used by the IoT device to compress data before queuing in the buffer. As for its transmission scheme, it chooses the maximum number of data packets to be sent in each timeslot (subject to battery availability).

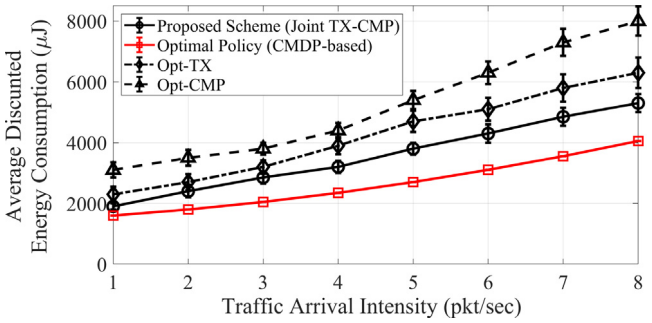

Fig. 7. Impact of sensor events arrival rate on average discounted energy consumption.

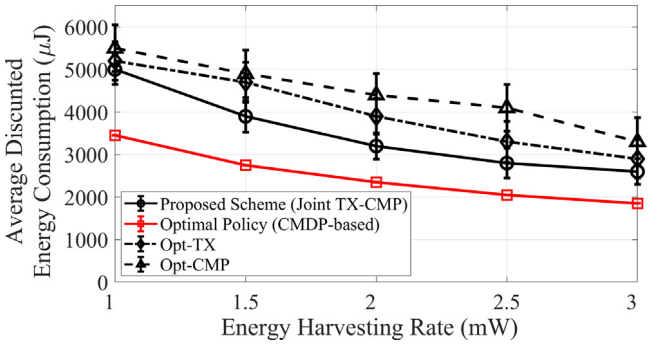

Fig. 8. Impact of the harvesting power on the average energy consumption.

- **Optimal Transmission (OPT-TX) policy:** It solely controls the transmission module of the IoT device in adaptation to the time-varying states of the environment. In other words, it performs no compression on the arriving data.

The operating policy for both OPT-CMP and OPT-TX is computed in a model-free fashion in the sense that their near-optimal policy has been computed with a similar constrained model-free scheme as with our own algorithm. Also, both policies satisfy the constraint on the mean buffer length, and aim at minimizing the long-term average discounted energy consumption. As such, in all the following experiments, the comparisons are made only in terms of the average discounted energy consumption $\bar{C}_E^{\pi}$. In fact, $\bar{C}_E^{\pi}$ is minimum in an optimal algorithm while the average buffer length stays below the threshold δ.

We have executed all four algorithms with identical configurations to evaluate their performance. Also, we investigate the impact of the system parameters on the achievable performance in three scenarios: (1) varying sensor data arrival rate, (2) different harvesting powers, and (3) different energy buffer capacity. Each point in the following plots is the average of 250 simulation runs. We include error bars which indicate 95% confidence that the actual average is within the range of depicted interval.

5.3.1. The impact of sensor data arrival rate on average energy consumption

To investigate the impact of sensor data arrival rate on the average energy consumption, all simulation settings are the same as those given in Table 2. However, we specify the constraint on average buffer length as $\delta = 8.25$, and vary the data arrival rates from one to eight data packets in each timeslot. As expected, when the arrival rate of events increases, the average discounted energy consumption increases as well. As shown in Fig. 7, this increase exhibits an almost linear trend. As seen in the figure, the proposed (Joint TX-CMP) algorithm consumes less energy in comparison with the other two suboptimal algorithms, and shows better performance in the long-term.

5.3.2. The impact of harvesting power on average energy consumption

To study the impact of the harvesting power on the average buffer length constraint, we consider a simulation setup where $\delta = 8.25$. We vary the harvesting power $\mathcal{P}$ from 1 to 3 mW during active states. As shown in Fig. 8, with the increase in the harvesting power, the total energy consumption is reduced. In fact, under a fixed buffer occupancy constraint, the more energy is available in the super-capacitor, the higher will become the number of feasible actions in different states. Hence, the controller would be able to better utilize its ample energy resources to exploit favorable channel opportunities and come up with a more judicious consumption plan in which the constraint is always met while avoiding inefficient energy consumption. It is also noticeable that under a relatively high harvesting power, the difference between the two baselines and our scheme would become smaller.

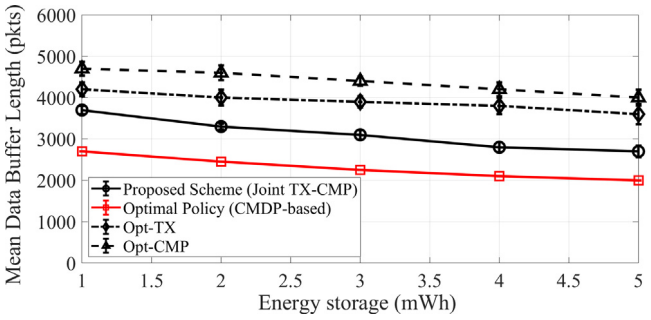

Fig. 9. Impact of energy buffer capacity on the average discounted energy consumption.

5.3.3. The impact of energy buffer capacity on average energy consumption

To assess the effect of energy buffer capacity on the average energy consumption, we use a setting where the constraint on mean data buffer length δ is set to 4.15. The energy buffer capacity is varied from 1 to 5 mWh. Ultra-capacitors with capacities 1, 2, 4, and 5 (mWh) are manufactured by several companies such as Tecate Group (e.g., see [52] for TPL-1.0f, TPL-2.0f, TPL-4.0f, TPL-5.0f models). As illustrated in Fig. 9, the increase in energy buffer capacity results in reduction in the average discounted energy consumption. In particular, similarly to the case of increasing the harvesting power, by also increasing the capacity for storing more harvested energy from the environment, the IoT node can better exploit the channel conditions. In other terms, to satisfy the data buffer constraint, it opportunistically sends a larger number of packets when the channel is good (thanks to the higher available energy), practically emptying its data buffer. Conversely, under poor channel conditions, it rarely transmits, or does not transmit at all, without having to worry about violating the data buffer constraint. However, as it is noticeable from the plot in Fig. 9, the reduction in energy usage occurs with a steeper slope when the harvesting power increases.

It can be concluded from Figs. 7–9 that under all configurations, while the proposed scheme alongside Opt-CMP and Opt-TX have successfully satisfied the buffer constraint, our Joint TX-CMP algorithm outperforms the other two due to having less average discounted energy consumption. This superiority lies in the fact that the proposed algorithm makes foresighted decisions about both the compression level as well as data transmission rate, effectively considering the subsequent impacts the current decisions may have on the future performance of the system. In contrast, the two baseline schemes act myopically with respect to either one of the control knobs, considering only the instantaneous performance, with no regard for the future trajectory of the system.

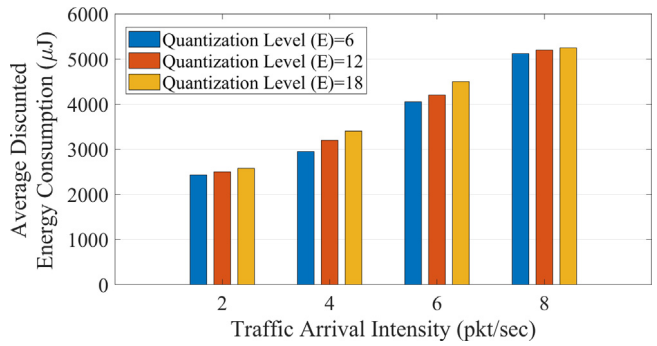

Fig. 10. Impact of quantization levels on the average discounted energy consumption.

5.4. The impact of quantization levels on average energy consumption

In one last experiment, in Fig. 10, we investigate the impact of the accuracy with which we quantize the energy storage space. Obviously, as the number of quantization levels increases, our quantized state space model would become a more accurate representative of the originally continuous space. A similar observation is expected for the three baseline schemes since their underlying state space model is the same. The plot in Fig. 10 also shows how the accuracy of the energy space model can affect the performance of the IoT node under varying load conditions. A notable observation is the negligibility of the differences under the lowest and highest load regimes. Under these two regimes, the battery state effectively remains within a certain range with a high probability, and a more accurate quantization would only complicate the system model without making much difference.

6. Conclusion and outlook

In this paper, the problem of jointly controlling the compression level and the transmission rate was studied for an IoT node equipped with a renewable energy source. Given the uncertainties posed by the randomness of the arrival processes of energy and sensory events, as well as the variations in channel qualities, the problem was formulated as a Constrained Markov Decision Process with the objective of minimizing the long-term average energy consumption, while satisfying an average delay constraint for the sensor events buffered as data packets in the node. To solve the optimization problem, we proposed a model-free reinforcement learning algorithm which computes a control policy in adaptation to the dynamics in the environment. Our algorithm operates in a model-free fashion in that it does not need prior statistical knowledge of the random processes in the environment. We may yet extend our solution to a multi-hop IoT network to consider scenarios where several nodes are connected and collaborate with each other to report their sensing data. This scenario would technically correspond to a multi-agent setting in which multiple nodes operate together in a multi-state environment. The formalization of such problems needs a decentralized MDP (DEC-MDP) [53] or a Markov game [54] formulation to also explicitly capture the interplay between the decisions of multiple agents. In fact, as the output from one agent (IoT device) would affect the input to the other agent, our single-agent formulation would no longer be valid in principle. The systematic extension of our scheme to this new setup would not be trivial as one has to deal with the computation of a globally optimal or (equilibrium) policy under very high dimensional joint state and action spaces and across the entire network. We believe, however, this can constitute a very interesting direction for future extension.

CRediT authorship contribution statement

Vesal Hakami: Conceptualization, Formal analysis, Project administration, Supervision, Validation, Writing - review & editing. **Seyedakbar Mostafavi:** Data curation, Investigation, Software, Visualization, Writing - review & editing. **Nastooh Taheri Javan:** Data curation, Visualization. **Zahra Rashidi:** Investigation, Writing - original draft, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] C. Pielli, A. Biason, A. Zanella, M. Zorzi, Joint optimization of energy efficiency and data compression in TDMA-based medium access control for the IoT, in: Proceedings of the IEEE Globecom Workshops (GC Wkshps), 2016, pp. 1–6.
[2] H.-S. Liu, C.-C. Chuang, C.-C. Lin, R.-I. Chang, C.-H. Wang, C.-C. Hsieh, Data compression for energy efficient communication on ubiquitous sensor networks, Tamkang J. Sci. Eng. 14 (3) (2011) 245–254.
[3] M. Vecchio, R. Giaffreda, F. Marcelloni, Adaptive lossless entropy compressors for tiny IoT devices, IEEE Trans. Wireless Commun. 13 (2) (2014) 1088–1100.
[4] Ukil, S. Bandyopadhyay, A. Pal, IoT data compression: sensor-agnostic approach, in: Proceedings of the Data Compression Conference, 2015, pp. 303–312.
[5] B. Tavli, I.E. Bagci, O. Ceylan, Optimal data compression and forwarding in wireless sensor networks, IEEE Commun. Lett. 14 (5) (2010) 408–410.
[6] M.J. Neely, Dynamic data compression for wireless transmission over a fading channel, in: Proceedings of the 42nd Annual Conference on Information Sciences and Systems, 2008, pp. 1210–1215.
[7] M. Centenaro, M. Rossi, M. Zorzi, Joint optimization of lossy compression and transport in wireless sensor networks, in: 2016 IEEE Globecom Workshops (GC Wkshps), Washington, DC, 2016, pp. 1–6.
[8] M.J. Neely, A. Sharma, Dynamic data compression with distortion constraints for wireless transmission over a fading channel, 2008, arXiv preprint arXiv: 0807.3768.
[9] C. Pielli, Č. Stefanović, P. Popovski, M. Zorzi, Joint compression, channel coding, and retransmission for data fidelity with energy harvesting, IEEE Trans. Commun. 66 (4) (2018) 1425–1439.
[10] Y. Yu, B. Krishnamachari, V.K. Prasanna, Data gathering with tunable compression in sensor networks, IEEE Trans. Parallel Distrib. Syst. 19 (2) (2008) 276–287.
[11] M. Vecchio, R. Giaffreda, F. Marcelloni, Adaptive lossless entropy compressors for tiny IoT devices, IEEE Trans. Wireless Commun. 13 (2) (2014) 1088–1100.
[12] M. Tahir, R. Farrell, Optimal communication-computation tradeoff for wireless multimedia sensor network lifetime maximization, in: 2009 IEEE Wireless Communications and Networking Conference, Budapest, 2009, pp. 1–6.
[13] W. Hu, W. Zhang, H. Hu, Y. Wen, K. Tseng, Toward joint compression–transmission optimization for green wearable devices: An energy-delay tradeoff, IEEE Internet Things J. 4 (4) (2017) 1006–1018.
[14] D. Incebacak, R. Zilan, B. Tavli, J.M. Barcelo-Ordinas, J. Garcia-Vidal, Optimal data compression for lifetime maximization in wireless sensor networks operating in stealth mode, Ad Hoc Netw. 24 (2015) 134–147.
[15] K.C. Barr, K. Asanović, Energy-aware lossless data compression, ACM Trans. Comput. Syst. 24 (3) (2006) 250–291.
[16] R. Sharma, A data compression application for wireless sensor networks using LTC algorithm, in: Proceedings of the IEEE International Conference on Electro Information Technology, 2015, pp. 598–604.
[17] S. Eswaran, J. Edwards, A. Misra, T.F.L. Porta, Adaptive in-network processing for bandwidth and energy constrained mission-oriented multihop wireless networks, IEEE Trans. Mob. Comput. 11 (9) (2012) 1484–1498.
[18] M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, Layering as optimization decomposition: A mathematical theory of network architectures, Proc. IEEE 95 (1) (2007) 255–312.
[19] W. Zhang, R. Fan, Y. Wen, F. Liu, Energy optimal wireless data transmission for wearable devices: A compression approach, IEEE Trans. Veh. Technol. 67 (10) (2018) 9605–9618.

[20] Min Jae Kang, et al., Energy-aware determination of compression for low latency in solar-powered wireless sensor networks, Int. J. Distrib. Sens. Netw. (2017).
[21] M.I. Mohamed, W.Y. Wu, M. Moniri, Adaptive data compression for energy harvesting wireless sensor nodes, in: 2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC), Evry, 2013, pp. 633–638.
[22] M.L. Putterman, Markov Decision Processes.: Discrete Stochastic Dynamic Programming, John Wiley and Sons, 2014.
[23] P. Castiglione, O. Simeone, E. Erkip, T. Zemen, Energy management policies for energy-neutral source-channel coding, IEEE Trans. Commun. 60 (9) (2012) 2668–2678.
[24] Michael Neely, Stochastic Network Optimization with Application to Communication and Queueing Systems, Morgan & Claypool, 2010.
[25] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[26] Abhishek B. Sharma, Leana Golubchik, Ramesh Govindan, Michael J. Neely, Dynamic data compression in multi-hop wireless networks, in: Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '09), ACM, New York, NY, USA, pp. 145–156.
[27] W. Hu, W. Zhang, H. Hu, Y. Wen, K. Tseng, Toward joint compression–transmission optimization for green wearable devices: An energy-delay tradeoff, IEEE Internet Things J. 4 (4) (2017) 1006–1018.
[28] C. Tapparello, O. Simeone, M. Rossi, Dynamic compression-transmission for energy-harvesting multihop networks with correlated sources, IEEE / ACM Trans. Netw. 22 (6) (2014) 1729–1741.
[29] Y. Cui, V.K.N. Lau, R. Wang, H. Huang, S. Zhang, A survey on delay-aware resource control for wireless systems—Large deviation theory, stochastic Lyapunov drift, and distributed stochastic learning, IEEE Trans. Inform. Theory 58 (3) (2012) 1677–1701.
[30] D. Zordan, T. Melodia, M. Rossi, On the design of temporal compression strategies for energy harvesting sensor networks, IEEE Trans. Wireless Commun. 15 (2) (2016) 1336–1352.
[31] E. Altman, Constrained Markov Decision Processes, CRC Press, 1999.
[32] C.J.C.H. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (3–4) (1992) 279–292.
[33] V.S. Borkar, Stochastic approximation with two-time scales, Systems Control Lett. 29 (5) (1997) 291–294.
[34] H. Wang, N. Mandayam, A simple packet transmission scheme for wireless data over fading channels, IEEE Trans. Commun. 52 (7) (2004) 1055–1059.
[35] C. Tan, N. Beaulieu, On first-order Markov modeling for the Rayleigh fading channel, IEEE Trans. Commun. 48 (12) (2000) 2032–2040.
[36] R.A. Berry, R.G. Gallager, Communication over fading channels with delay constraints, IEEE Trans. Inform. Theory 48 (5) (2002) 1135–1149.
[37] C.K. Ho, P.D. Khoa, P.C. Ming, Markovian models for harvested energy in wireless communications, in: Proc. 2010 IEEE International Conf. Commun. Syst., pp. 311–315.
[38] B. Medepally, N.B. Mehta, C.R. Murthy, Implications of energy profile and storage on energy harvesting sensor link performance, in: Proc. 2009 IEEE Global Telecommun. Conf..
[39] J. Lei, R. Yates, L. Greenstein, A generic model for optimizing single-hop transmission policy of replenishable sensors, IEEE Trans. Wirel. Commun. 8 (2) (2009) 547–551.
[40] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.
[41] R. Aslani, V. Hakami, M. Dehghan, A token-based incentive mechanism for video streaming applications in peer-to-peer networks, Multimedia Tools Appl. 77 (12) (2018) 14625–14653.
[42] Joelle Pineau, Geoff Gordon, Sebastian Thrun, Point-based value iteration: an anytime algorithm for POMDPs, in: Proceedings of the 18th International Joint Conference on Artificial intelligence (IJCAI'03), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1025–1030.
[43] M.G. Lagoudakis, R. Parr, Least squares policy iteration, J. Mach. Learn. Res. 4 (2003) 1107–1149.
[44] T.M. Mitchell, Machine Learning, ed: McGraw-Hill, Boston, MA, 1997.
[45] Shige Peng, Stochastic Hamilton–Jacobi-Bellman equations, SIAM J. Control Optim. 30 (2) (1992) 284–304.
[46] V.S. Borkar, An actor-critic algorithm for constrained Markov decision processes, Systems Control Lett. 54 (3) (2005) 207–213.
[47] NESSCAP Energy Inc., NESSCAP ultracapacitor products. Available: www.nesscap.com.
[48] IEEE Std 802.15.4-2006, IEEE standard for information technology— telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications, 2006.
[49] Q. Zhang, S. Kassam, Finite-state markov model for rayleigh fading channels, IEEE Trans. Commun. 47 (11) (1999) 1688–1692.
[50] H.S. Wang, N. Moayeri, Finite-state markov channel-a useful model for radio communication channels, IEEE Trans. Veh. Technol. 44 (1) (1995) 163–171.
[51] Crossbow Technology Inc., MICAz: wireless measurement system. Available: www.xbow.com.
[52] Tecate Group, General Purpose Radial Lead Ultracapacitor Cells, Available: https://www.capcomp.de/fileadmin/Webdata/partner/TECATE/Technical_Data/Ultracap/TEC_2016_TPL.pdf.
[53] D.S. Bernstein, R. Givan, N. Immerman, S. Zilberstein, The complexity of decentralized control of Markov decision processes, Math. Oper. Res. 27 (4) (2002) 819–840.
[54] L.S. Shapley, Stochastic games, Proc. Natl. Acad. Sci. 39 (10) (1953) 1095–1100.