12-2015

Adaptive duty cycling in sensor networks with energy harvesting using continuous-time markov chain and fluid models

Ronald Wai Hong Chan

Pengfei Zhang

Ido Nevat

Sai Ganesh Nagarajan

Alvin Cerdena VALERA
Singapore Management University, alvinvalera@smu.edu.sg

See next page for additional authors

DOI: https://doi.org/10.1109/JSAC.2015.2478717

Author
Ronald Wai Hong Chan, Pengfei Zhang, Ido Nevat, Sai Ganesh Nagarajan, Alvin Cerdena VALERA, and Hwee Xian TAN

Published in IEEE Journal on Selected Areas in Communications, 2015 December, Volume 33, Issue 12, Pages 2687-2700
https://doi.org/10.1109/JSAC.2015.2478717

Adaptive Duty Cycling in Sensor Networks With Energy Harvesting Using Continuous-Time Markov Chain and Fluid Models

Wai Hong Ronald Chan, Pengfei Zhang, Ido Nevat, Sai Ganesh Nagarajan, Alvin C. Valera, Hwee-Xian Tan, and Natarajan Gautam

***Abstract*—The dynamic and unpredictable nature of energy harvesting sources available for wireless sensor networks, and the time variation in network statistics like packet transmission rates and link qualities, necessitate the use of adaptive duty cycling techniques. Such adaptive control allows sensor nodes to achieve long-run energy neutrality, where energy supply and demand are balanced in a dynamic environment such that the nodes function continuously. In this paper, we develop a new framework enabling an adaptive duty cycling scheme for sensor networks that takes into account the node battery level, ambient energy that can be harvested, and application-level QoS requirements. We model the system as a Markov decision process (MDP) that modifies its state transition policy using reinforcement learning. The MDP uses continuous time Markov chains (CTMCs) to model the network state of a node to obtain key QoS metrics like latency, loss probability, and power consumption, as well as to model the node battery level taking into account physically feasible rates of change. We show that with an appropriate choice of the reward function for the MDP, as well as a suitable learning rate, exploitation probability, and discount factor, the need to maintain minimum QoS levels for optimal network performance can be balanced with the need to promote the maintenance of a finite battery level to ensure node operability. Extensive simulation results show the benefit of our algorithm for different reward functions and parameters.**

***Index Terms*—Wireless sensor networks, adaptive duty cycle, continuous-time Markov chain, Markov decision process, reinforcement learning, fluid model.**

I. Introduction

WIRELESS sensor networks (WSNs) can be used in a large number of applications, such as environmental and structural health monitoring, weather forecasting [1]–[3], surveillance, health care, and home automation [4]. A key challenge that constrains the operation of sensor networks is limited lifetime arising from the finite energy storage in each node [5]. However, recent advances in energy harvesting technologies are enabling the deployment of sensor nodes that are equipped with a replenishable supply of energy [6]–[10]. These techniques can potentially eliminate the limited lifetime problem in sensor networks and enable perpetual operation without the need for battery replacement, which is not only labourious and expensive, but also infeasible in certain situations.

Despite this, the uninterrupted operation of energy harvesting-powered wireless sensor networks (EH-WSNs) remains a major challenge, due to the unpredictable and dynamic nature of the harvestable energy supply [5], [11]. To cope with the energy supply dynamics, adaptive duty cycling techniques [11]–[17] have been proposed. The common underlying objective of these techniques is to attain an optimal *energy-neutral* point at every node, wherein the energy supply and energy demand are balanced. Also, other works focus on optimizing energy consumption in EH-WSNs by formulating the response to a time-varying harvesting profile as a Markov Decision Process (MDP) or other probability-driven processes [18], [19]. These energy-oriented techniques tend to focus primarily on obtaining the optimal per-node duty cycle to prolong network lifetime, while neglecting application-level quality of service (QoS) requirements [20]–[22]. More recently, adaptive duty cycling techniques involving MDPs have been proposed that focus on achieving energy efficient operations while considering a subset of the QoS requirements (such as throughput or delay) with full channel state information [23]–[27], but without considering the long-term energy availability of the system or tolerating ambiguity in the state information.

In this paper, we develop a novel framework enabling an adaptive duty cycling scheme that allows network designers to trade-off between both short-term QoS requirements and long-term energy availability using an adaptive reinforcement learning algorithm. In our framework, the QoS metrics of the system are estimated based on knowledge of the average performance of the network, such as the average packet transmission and probing rates, without necessarily requiring knowledge of the full channel state information. These quantities can be monitored online or estimated offline with a time delay depending on the requirements of the system. Fig. 1 illustrates the main components of such a scheme, which comprises the following: (i) energy harvesting controller; (ii) adaptive duty cycle controller; and (iii) wakeup scheduler.

The energy harvesting controller provides information on the amount of harvested energy that is currently available,

Manuscript received March 29, 2015; revised July 11, 2015; accepted September 14, 2015. Date of publication September 14, 2015; date of current version November 16, 2015.

W. H. R. Chan is with the Mechanical Engineering Department, Stanford University, Stanford, CA 94305 USA.

P. Zhang, I. Nevat, and S. G. Nagarajan are with the Institute for Infocomm Research (I^2R), Singapore 138632, Singapore.

A. Valera and H.-X. Tan are with the School of Information Systems, Singapore Management University (SMU), Singapore 178902, Singapore.

N. Gautam is with the Industrial and Systems Engineering Department, College Station, Texas A&M University, Texas 77843 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2015.2478717

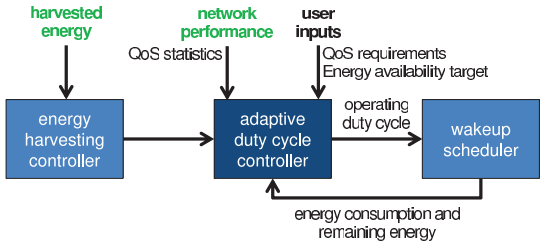


Fig. 1. Main components of proposed adaptive scheme. Quantities that fluctuate due to environmental influences are marked in green.

while predicting the harvested energy available within the next few hours, depending on the diurnal cycles of the energy sources. The adaptive duty cycle controller computes the optimal operating duty cycle based on user inputs (in the form of application QoS requirements) and the available amount of harvested energy. The wakeup scheduler will then: (i) manage the sleep and wake interfaces of each node, based on the recommended operating duty cycle, and (ii) provide feedback to the adaptive duty cycle controller on the energy consumption of and remaining energy in the node. This feedback loop allows the duty cycle controller to **adapt** its duty cycle, based on the harvested and remaining energies - in order to meet QoS requirements - via operating policies such as energy neutrality.

In our previous work [28], we developed a duty cycle controller key to the energy-aware operations of a sensor network. Using a Continuous Time Markov Chain (CTMC) model, we derived key QoS metrics including *loss probability*, *latency*, as well as *power consumption*, as functions of the duty cycle. (We define these metrics more explicitly in Section III-B.) We then formulated and solved the optimal operating duty cycle as a non-linear optimization problem, using latency and loss probability as the constraints. We validated our CTMC model through Monte Carlo simulations and demonstrated that a Markovian duty cycling scheme can outperform periodic duty cycling schemes. In this paper, we extend the previous work and enhance the duty cycle controller by considering the battery level and energy harvesting rate. We then formulate the adaptive duty cycle problem as a MDP model. The states of the MDP model correspond to the energy consumption rates at which the node can operate. The actions refer to the transition rates between the various duty cycle values that the node can adopt. The reward function is derived based on the QoS parameters derived in the previous paper, as well as on the energy availability of the battery based on a fluid model [29]–[31] that indicates the ability of the node to function continuously. While finding the optimal duty cycle scheme to operate under is a nonconvex optimization problem which is hard to implement on-line, we propose a relatively simple on-line approach which uses reinforcement learning [32], [33] to heuristically update the reward function to approximate convergence. We also use extensive simulations to show that the MDP converges to a desirable result quickly, and to compare our approach with a random approach to demonstrate the performance of the MDP scheme.

In this work, we make several key contributions: (i) we enable a WSN to determine its duty cycle control through simultaneous consideration of the energy supply dynamics and application-level QoS requirements; (ii) we establish a reward framework that allows the network to tune the relative importance of the energy availability and the QoS requirements; (iii) we implement a reinforcement learning algorithm that converges to a desirable solution quickly and with lower computational complexity than convergence to a fully optimal solution; and (iv) we allow the system to adapt to changes in the environment and/or the network that occur at timescales larger than the convergence time of the learning curve. In Fig. 2, we provide an approximate comparison of the timescales involved in our system.

The rest of the paper is organized as follows. Section II provides details on the key assumptions used in the system model for a battery-free framework. In Section III, we derive network performance metrics using a CTMC model. We extend the system model to incorporate battery levels in Section IV, and describe the behaviour of the battery with another CTMC model in Section V. In Section VI, we allow the system to determine the optimal rates of transition between its constituent states using a MDP model. Simulation results are presented in Section VII. Section VIII concludes the paper.

II. System Model for Battery-Free Framework

In this Section, we develop a probabilistic model that describes the features of a single WSN node, i.e. data reception from other nodes and data transmission towards the gateway (GW) via another node. Both this node and its recipient node are duty cycled, and several QoS parameters are investigated as functions of this duty cycle. Under the framework developed in this Section, we do not consider the role of energy harvesting. Thus, we assume the network is powered by mains electricity, and the effective battery capacity of each node is unlimited (although we would still try to minimise the energy consumption of each node). In Section IV, we will generalise our framework by considering energy harvesting nodes.

We now present all the statistical assumptions of our framework and provide details of various system components required, such as the traffic model, channel model and packet transmission schemes.

1) Node State: Each node v_j is in one of the following states $N_j \in \{0, 1\}$ at any point in time, where $N_j = 0$ and $N_j = 1$ denote that v_j is in the *asleep* and *awake* states respectively. The duration t that node v_j is in each of the states N_j is a random variable that follows an exponential distribution:

$$p(t) = \begin{cases} \gamma_i \cdot e^{-\gamma_i \cdot t} & t \geq 0 \\ 0 & t < 0, \end{cases} \quad (1)$$

where γ_i, $i \in \{0, 1\}$ are the rates of the asleep and awake states. The average long-term fraction of time that the node is *awake* is given by $q = \frac{1}{\gamma_1 \cdot T}$ where $T = \frac{1}{\gamma_0} + \frac{1}{\gamma_1}$ is the average cycle time.

2) Traffic Model: The number of data packets d_0 generated by each node follows a Poisson distribution with an average rate of λ_0 packets per unit time, i.e., $d_0 \sim \text{Pois}(\lambda_0)$.

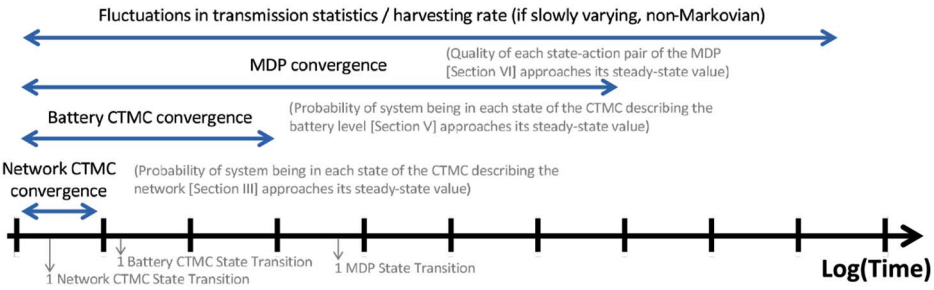


Fig. 2. Comparison of timescales involved in our system.

In addition, the node receives d_n packets from all of its neighbours according to a Poisson process $d_n \sim \text{Pois}(\lambda)$.

3) Wireless Channel Model: The time-varying wireless link quality is modelled by the classical Gilbert-Elliot Markovian model [34], [35] with two states $L \in \{0, 1\}$, where $L = 0$ and $L = 1$ denote that the channel quality is *bad* and *good* respectively. The duration t that a node is in each of the channel states is a random variable that follows an exponential distribution:

$$p(t) = \begin{cases} c_i \cdot e^{-c_i \cdot t} & t \geq 0 \\ 0 & t < 0, \end{cases} \tag{2}$$

where $c_i, i \in \{0, 1\}$ are the respective rates of the *bad* and *good* states. We let β and α denote the probabilities of successfully delivered data packets when the channel is in the *bad* and *good* states respectively. Acknowledgment packets are assumed to always be delivered successfully.

4) Probing Mechanism: The network utilizes probes to determine if an arbitrary downstream node v_k is in an awake state $N_k = 1$, prior to the commencement of data transmission. The probing mechanism is modelled as a Poisson process, with intensities θ_g and θ_b when the channel quality is good and bad respectively. The reception of a probe-acknowledgment by the transmitter node v_j indicates that v_k is awake; v_j will then instantaneously transmit *all* its data packets to v_k.

5) Transmission Schemes: We consider two transmission schemes:
   a) **No Retransmissions**: data packets that have not been successfully delivered to the receiver (due to poor channel quality) will **not** be retransmitted. The corresponding average numbers of packets that successfully arrive at a node under good and poor channel conditions are denoted as λ_g and λ_b respectively, where $\lambda_b = \frac{\beta \cdot \lambda_g}{\alpha}$. We denote this scheme by $\mathcal{X}_n$.
   b) **Retransmissions**: data packets are retransmitted until they are successfully delivered to the receiver. The corresponding average number of packets that successfully arrive under this scheme is λ, where $\lambda_g = \lambda_b = \lambda$. The effective packet arrival rate when the node is in the awake state is $\frac{\lambda}{q}$. We denote this scheme by $\mathcal{X}_r$.

6) Power Consumption Parameters: The power consumptions of a node in the asleep and awake states are $\mathcal{P}_{asleep}$ and $\mathcal{P}_{awake}$ respectively. The power consumption of the probing mechanisms is denoted as $\mathcal{P}_{probe}$. The energy incurred to transmit a single data packet is $\mathcal{E}_{tx}$.

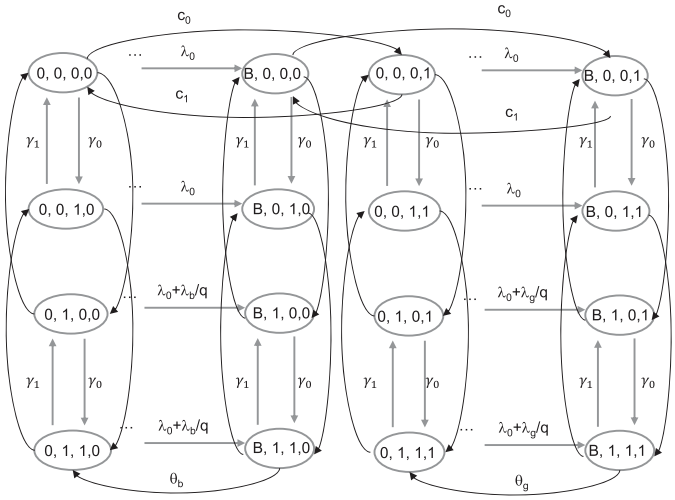


Fig. 3. CTMC model of a two-node section of a network. The CTMC is guided by the transition rate matrix $\mathcal{Q}$.

With these definitions, we now present in the next Section a probabilistic model based on a Continuous Time Markov Chain (CTMC) model, and derive the QoS parameters of interest.

III. CTMC Model For Node Performance State (Battery-Free; Fixed Duty Cycle)

In this Section, we design a probabilistic model to describe the performance of a single node in the WSN for a fixed duty cycle q. To this end, we model the system as a CTMC, as shown in Fig. 3. In addition, we assume that the system evolves independently of the battery level, as we will be using this model to generate instantaneous QoS metrics for the system, assuming in addition that the timescale of energy fluctuations exceeds the timescale of packet traffic equilibration (i.e. the convergence time of this CTMC).

A. CTMC State Space Model

We consider a 4-tuple CTMC state space as follows:

1) Node buffer: Each node has a FIFO buffer of finite size B. The number of packets in the finite queue is denoted by $b \in \{0, 1, \ldots, B\}$.

2) Node state: As mentioned in Section II, each node v_j is in state $N_j \in \{0, 1\}$ at any one time, depending on whether it is *asleep* ($N_j = 0$) or *awake* ($N_j = 1$).
3) Downstream node state: An arbitrary downstream (receiving) node v_k is in state $N_k \in \{0, 1\}$ at any one time, depending on whether it is *asleep* ($N_k = 0$) or *awake* ($N_k = 1$).
4) Link quality: The wireless link quality is in state $L \in \{0, 1\}$ at any one time, depending on whether the channel is *bad* ($L = 0$) or *good* ($L = 1$). (We note that it is straightforward to extend the analysis to multiple link quality states.)

Given these definitions, the state space $\mathcal{S}$ can be written as the following Cartesian product $\mathcal{S} = \{0, 1, \ldots, B\} \times \{0, 1\} \times \{0, 1\} \times \{0, 1\} \in \mathcal{R}^{|b|\times|N_j|\times|N_k|\times|L|}$. The corresponding cardinality of the state space is given by $|\mathcal{S}| = 8(B + 1)$.

B. QoS metric definitions

We now present the following key QoS metrics of interest:

- *loss probability* due to wireless channel transmission errors and packet drops arising from buffer overflows, denoted $\pi\,(q)$;
- *latency* incurred by holding packets in the transmission queue, denoted $\ell\,(q)$;
- and *average power consumption* incurred by a node, denoted $\rho\,(q)$.

In the next Lemma we present the derived expressions for these QoS metrics for the two cases under consideration, namely the Retransmissions and No Retransmissions schemes.

Lemma 1: B. QoS metrics under Retransmissions scheme

The QoS parameters are given by (3), as shown at the bottom of the page.

Proof: See [28]. ■

Lemma 2: B. QoS metrics under No Retransmissions scheme

The QoS parameters are given by (4), as shown at the bottom of the page.

Proof: See [28]. ■

By defining parameters (e.g. packet arrival rates λ_0 and λ, probing intensities θ_g and θ_b, and maximum buffer size B) of the transitive matrix $\mathcal{Q}$ according to parameters of the actual sensor network, we can obtain the optimal duty cycle q in terms of the asleep and awake rates γ_0 and γ_1. This is presented next.

C. Optimal Duty Cycle

To find the optimal duty cycle, different criteria can be considered. Here, we choose to find the duty cycle which minimizes the power consumption, while satisfying application-level QoS constraints. We note that other criteria could be considered and our framework is general enough to handle them as well.

For our criterion, the resulting optimisation problem is given as follows:

$$q = \arg\min_{q} \rho(q) \qquad \text{subject to } \pi(q) \le \pi_0,\ \ell(q) \le \ell_0,\ q \ge 0 \tag{5}$$

where π_0 and ℓ_0 are pre-defined latency and loss thresholds.

Recall that γ_1 and γ_0 can be expressed as functions of q and the average cycle time T as follows:

$$\gamma_1 = \frac{1}{T \cdot q} \qquad\qquad \gamma_0 = \frac{1}{T \cdot (1 - q)} \tag{6}$$

Thus, we can further simplify the optimisation problem to a single parameter optimization problem by defining T and solving for γ_1 and γ_0. Hence, even though the optimization problem in (5) does not have an analytical closed form expression, it is

---

$$\pi\,(q) = \sum_{k=0}^{1} \frac{p_{B,1,0,k}(\lambda/q + \lambda_0)}{\lambda/q + \lambda_0 + \gamma_0 + \gamma_1 + c_k} + \sum_{k=0}^{1} \frac{p_{B,1,1,k}(\lambda/q + \lambda_0)}{\lambda/q + \lambda_0 + 2\gamma_1 + \theta + c_k} + \sum_{k=0}^{1} \frac{p_{B,0,0,k}\lambda_0}{\lambda_0 + 2\gamma_0 + c_k} + \sum_{k=0}^{1} \frac{p_{B,0,1,k}\lambda_0}{\lambda_0 + \gamma_0 + \gamma_1 + c_k}$$

$$\ell\,(q) = \frac{1}{\lambda + \lambda_0} \sum_{b=0}^{B} \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{k=0}^{1} b p_{b,i,j,k}$$

$$\rho\,(q) = \mathcal{P}_{asleep} \sum_{b=0}^{B} \sum_{j=0}^{1} \sum_{k=0}^{1} p_{b,0,j,k} + \mathcal{P}_{awake} \sum_{b=0}^{B} \sum_{j=0}^{1} \sum_{k=0}^{1} p_{b,1,j,k} + \mathcal{P}_{probe} \sum_{b=1}^{B} \sum_{j=0}^{1} \sum_{k=0}^{1} p_{b,1,j,k} + \left(\frac{\lambda}{1 - \beta/\alpha} + \lambda_0\right) \mathcal{E}_{tx} \tag{3}$$

$$\pi\,(q) = \frac{p_{B,1,0,0}(\lambda_b/q + \lambda_0)}{\lambda_b/q + \lambda_0 + \gamma_0 + \gamma_1 + c_0} + \frac{p_{B,1,1,0}(\lambda_b/q + \lambda_0)}{\lambda_b/q + \lambda_0 + 2\gamma_1 + \theta + c_0} + \frac{p_{B,0,0,0}\lambda_0}{\lambda_0 + 2\gamma_0 + c_0} + \frac{p_{B,0,1,0}\lambda_0}{\lambda_0 + \gamma_0 + \gamma_1 + c_0} +$$
$$\frac{p_{B,1,0,1}(\lambda_g/q + \lambda_0)}{\lambda_g/q + \lambda_0 + \gamma_0 + \gamma_1 + c_1} + \frac{p_{B,1,1,1}(\lambda_g/q + \lambda_0)}{\lambda_g/q + \lambda_0 + 2\gamma_0 + \gamma_1 + c_1} + \frac{p_{B,0,0,1}\lambda_0}{\lambda_0 + 2\gamma_0 + c_1} + \frac{p_{B,0,1,1}\lambda_0}{\lambda_0 + \gamma_0 + \gamma_1 + c_1}$$

$$\ell\,(q) = \frac{1}{\lambda_b + \lambda_0} \sum_{b=0}^{B} \sum_{i=0}^{1} \sum_{j=0}^{1} b p_{b,i,j,0} + \frac{1}{\lambda_g + \lambda_0} \sum_{b=0}^{B} \sum_{i=0}^{1} \sum_{j=0}^{1} b p_{b,i,j,1}$$

$$\rho\,(q) = \mathcal{P}_{asleep} \sum_{b=0}^{B} \sum_{j=0}^{1} \sum_{k=0}^{1} p_{b,0,j,k} + \mathcal{P}_{awake} \sum_{b=0}^{B} \sum_{j=0}^{1} \sum_{k=0}^{1} p_{b,1,j,k} + \mathcal{P}_{probe} \sum_{b=1}^{B} \sum_{j=0}^{1} \sum_{k=0}^{1} p_{b,1,j,k} + (\lambda + \lambda_0)\mathcal{E}_{tx} \tag{4}$$

easy to find the optimal q (denoted q^*) numerically via simple evaluation on a finely divided grid.

IV. System Model for Framework with Finite Battery Level and Energy Harvesting

In this Section and the next, we develop a framework that describes the evolution of the battery level of a single node with finite battery capacity and the capability to harvest energy from the environment. Under the constraints of this framework, we eventually identify a set of variables that can be designed to control the performance of the network, namely the transition rates that govern the transition of the battery state between different rates of energy harvesting and consumption. The consumption rates were previously derived in Section III. Note that the battery states in this Section and the next are distinct from the CTMC states of Sections II and III, which describe the network status and capacity of the node.

Here, we provide details of additional system components required in this framework, in particular the battery model and its time evolution.

1) Battery Model: The battery level $X(t)$ of the sensor node is treated in a continuous fashion, i.e. $X(t) \in \Re$ and $X(t) \in [0, C]$, where C is the capacity of the battery and $t \geq 0$.
2) Rate of Change of Battery Level: In reality, the rate of change of $X(t)$ can take any value in a continuous, bounded interval $[\dot{X}_{\min}, \dot{X}_{\max}]$ where $\dot{X}_{\min}$ and $\dot{X}_{\max}$ are the minimum and maximum possible rates determined by the battery chemistry, the maximum harvesting power available, and the maximum power consumption of the node.
   We model this with a *fluid model* by sampling mn possible rates within this interval, and defining the cardinality of the state space of the battery to be mn. Let $Z(t)$ be the state of the battery at time t. When $Z(t)$ is in some state $i \in T = \{1, 2, \ldots, mn\}$, the evolution of the process satisfies

$$\frac{dX(t)}{dt} = \begin{cases} \max(0, r_i) & \text{if } X(t) = 0, \\ r_i & \text{if } 0 < X(t) < C, \\ \min(0, r_i) & \text{if } X(t) = C, \end{cases} \tag{7}$$

   where r_i is the rate governing the process evolution corresponding to state i provided the battery is neither full nor empty. In this model, if the battery is full, the rate of change of $X(t)$ with respect to time cannot take positive values; if the battery is empty, the rate of change of $X(t)$ with respect to time cannot take negative values. This allows us to model the continuous nature of the battery level using a set of discrete states that best describe the time evolution of the battery level by characterizing the most common rates of change in the battery level.
   Physically, the change in the battery level $X(t)$ is driven by two processes: the harvesting of energy from the surroundings at rate h_k, and the consumption of energy by the system at rate u_l.
   a) Discretized Energy Harvesting Rate: In reality, the energy harvesting rate h_k can take any value between zero and the maximum physically possible harvesting rate $h_{\max}$. We model this by sampling m possible harvesting rates between zero and $h_{\max}$.
   b) Discretized Energy Consumption Rate: In reality, the energy consumption rate u_l can take any value between zero and the maximum physically possible consumption rate $u_{\max}$. We model this by sampling n possible consumption rates between zero and $u_{\max}$. As demonstrated in our previous paper [28], u_l can be modelled as a monotonic function of the duty cycle q.
   c) Discretized Rate of Change of Battery Level: Now, we set $k, l \in \mathfrak{Z}$, $k \in (0, m]$ and $l \in (0, n]$, and we define

$$r_{n(k-1)+l} = h_k - u_l, \tag{8}$$

   choosing $\{h_k\}$ and $\{u_l\}$ such that the resulting r_i are unique. Note $\dot{X}_{\min} = -u_{\max}$ and $\dot{X}_{\max} = h_{\max}$.

V. CTMC Model For Node Battery State (Variable Duty Cycle)

With the model described in the previous Section, we can now extend our framework so that it also captures the time evolution of the battery level of a single node in the network. Based on this extended framework, we can thereby describe the *energy availability* of the battery in terms of a probabilistic description of the amount of time the battery level is above zero, as well as a *transition rate matrix* that describes the transition of the system from one set of energy harvesting and consumption rates to another set. By isolating the transitions between different harvesting rates from the transitions between different consumption rates, we then separate the influences of the environment from a potential set of user inputs for system optimization.

A. Characteristics of CTMC describing transitions between battery states

Here, we introduce a CTMC that describes the transitions between the mn battery states. This allows us to concretely set up the battery model, as well as to generate the energy availability required for the reward function of the Markov decision process (MDP) to be presented in Section VI.

Suppose the CTMC has a generator matrix $\mathcal{Q}_e^{MDP} = [q_{e,ij}]$. Let us define a drift matrix

$$D = \begin{bmatrix} r_1 & & & \\ & r_2 & & 0 \\ & & \ddots & \\ 0 & & & r_{mn} \end{bmatrix}. \tag{9}$$

In addition, let

$$\pi_{ij}(t) = P_r(Z(t) = j | Z(0) = i), \quad i, j \in T \tag{10}$$

and

$$\pi_j = \lim_{t\to\infty} P_r(Z(t) = j|Z(0) = i), \quad i, j \in T. \tag{11}$$

In other words, $\pi_{ij}(t)$ is the probability that the battery is in state j at time t given that it was initially in state i, and π_j is the limit of $\pi_{ij}(t)$ as t goes to infinity, assuming the CTMC has a stationary distribution. As in the results of Section III, the steady-state probabilities π_j should then satisfy $\mathrm{p_e}\mathcal{Q}_e^{MDP} = 0$ and $\sum_{j\in T} \pi_j = 1$, where $\mathrm{p_e} = [\pi_1\ \pi_2\ \ldots\ \pi_{|T|}]$.

B. Energy availability

The probability that the node battery of the node contains a non-zero amount of energy is given by the limiting availability, A.

Lemma 3: The energy availability A is given by

$$A = 1 - F_1(0) - F_2(0) - \cdots - F_{mn}(0), \tag{12}$$

where $F_j(x) = \lim_{t\to\infty} F(t, x, j; y, i)$. $F(t, x, j; y, i)$ gives the cumulative transition probability that the battery level $X(t)$ is at most x at time t and that the battery is in state j, given that the battery was originally in state i with battery level y.

Proof: See Appendix A. ■

In order to increase A, we have to choose the entries of the grand transition matrix $\mathcal{Q}_e^{MDP}$ appropriately such that the stationary probabilities π_j give us the lowest possible values of $F_j(0)$.

C. Decomposing the CTMC into harvesting and consumption states

Since the harvesting and consumption processes are physically distinct, we can decompose our CTMC into two subchains: one involving the transition between different harvesting rates, and one involving the transition between different consumption rates.

If we assume that the transitions between the harvesting rates take place randomly and independently of the transitions between the consumption rates, for example as in direct solar radiation [36], [37], then we can define a stationary distribution $\mathrm{p_h}$ and a transition matrix $\mathcal{Q}_h^{MDP}$ for the harvesting states, and a stationary distribution $\mathrm{p_u}$ and a transition matrix $\mathcal{Q}_u^{MDP}$ for the consumption states. We can then implement a Markovian scheme to transition between the various consumption rates. Note that $\mathrm{p_h}\mathcal{Q}_h^{MDP} = 0$, $\sum_{k\in\mathfrak{X},k\in(0,m]} \mathrm{p_h}(k) = 1$, $\mathrm{p_u}\mathcal{Q}_u^{MDP} = 0$ and $\sum_{l\in\mathfrak{X},l\in(0,n]} \mathrm{p_u}(l) = 1$. Then, $\mathrm{p_e}$ is the Kronecker product of the two component stationary distributions $\mathrm{p_h} \otimes \mathrm{p_u}$, while $\mathcal{Q}_e^{MDP} \equiv \mathcal{Q}_h^{MDP} \otimes \mathcal{Q}_u^{MDP}$.

Conversely, if we assume that the harvesting rate changes smoothly and periodically, for example as in diffuse solar radiation [38], then we should instead take $\mathcal{Q}_e^{MDP} \equiv \mathcal{Q}_u^{MDP}$ and assume the harvesting rate is sufficiently stationary that we ignore the harvesting transitions in the battery model CTMC. We then measure the harvesting rate h_k regularly and implement the MDP to be discussed in Section VI such that r_i is updated regularly. If the convergence rate of the MDP is faster than the timescale of the variation of the harvesting rate, then this method will be able to reasonably adapt to a changing harvesting rate.

1) Derivation of the harvesting transition matrix (for first assumption): $\mathcal{Q}_h^{MDP}$ and $\mathrm{p_h}$ can be derived from empirical data. For a solar-harvesting sensor node, one could measure the time variation of solar energy over a suitably long period of time, and then fit to the averaged data a CTMC whose statistics match the empirical distribution of solar energy throughout the day [36], [37].

2) Definition of the consumption transition matrix (for both assumptions): $\mathcal{Q}_u^{MDP}$ and $\mathrm{p_u}$ are user-defined inputs. In the MDP formulation to follow, we generate an adaptive transition matrix $\mathcal{Q}_u^{MDP}$ based on the optimization of the quality matrix $Q_m = [Q_{m,ls}]$ to be defined in Section VI-A.

VI. MDP Model For Variable Duty Cycle

With the model described in Sections II and III, we can quantify the performance of our system for a known duty cycle q assuming a sufficiently stationary battery level X such that the QoS metrics obtained can be approximated as instantaneous. Using this information, we can now construct a duty cycle policy that allows our system to respond to environmental variations, such as the sunlight available to a solar-harvesting node, while cognizant of the QoS targets that the system is required to fulfil. The effectiveness of the policy is measured both by the QoS targets, for which a model was provided in Sections II and III, and by the battery level, for which a model was provided in Sections IV and V. In addition, the policy is used to determine a good set of transition rates between the various consumption rates described in Section V-C.

A. MDP for Variable Duty Cycle with Reinforcement Learning

In principle, we could try out every single possibility of q for every possible harvesting rate h_k and consumption rate u_l to determine the best q for each rate of change of the battery level r_i. However, this is likely to be costly in terms of both time and computational resources. By formulating our problem as a Markov Decision Process (MDP) driven by a reinforcement learning algorithm, we could possibly reduce the number of computations, perform them online instead of offline, and make our system adaptive to changing system parameters, such as the instantaneous packet transmission and probing rates.

A MDP is useful to describe a decision making process that allows the system to transit between a set of *states*. MDPs have been used in various works to control duty cycling and channel usage in WSNs [39]–[43]. The decision maker has a set of *actions* that can be chosen to describe the state transitions. Each state transition is associated with an immediate scalar *reward*, which is given to the decision-maker or learner. Here, the goal of the reinforcement learning algorithm is to take actions, transit from one state to another, and maximize the expected sum of the rewards in the long run. To keep track of its rewards, the system maintains a *quality* function for each state-action pair, which is a cumulative measure of the rewards obtained so far, and consults this to take an action (with greedy

probability ϵ_t). Thus, by taking actions, obtaining rewards and updating the quality matrix based on this reward, the learner finally converges to a policy which approaches maximum return of rewards.

1) States: We define the set $\mathcal{F}$ of states f_l where $l \in \mathfrak{Z}$ and $l \in (0, n]$ such that each state corresponds to a unique consumption rate u_l.
2) Actions: We define the set $\mathcal{G}$ of actions g_s such that each action corresponds to a CTMC transition matrix $\mathcal{Q}_{u,s}^{MDP}$ corresponding to transitions between different discretized energy consumption rates. The action space can be designed such that each $\mathcal{Q}_{u,s}^{MDP}$ has a different stationary distribution. Different actions will then signify different duty cycle probability distributions in the limit of infinite time. For example, we could design an action that constrains the duty cycle to frequently take a low value, and another action that constraints it to frequently take a high value. The size of the square matrix $\mathcal{Q}_{u,s}^{MDP}$ is n. Here, we sample the entire space of possible $\mathcal{Q}_{u}^{MDP}$ to obtain a representative set of transition matrices that span the space and are physically convenient to implement.
3) Rewards: Based on the requirements of the user, be it a need to conserve energy aggressively, to consume energy aggressively, or to maintain a minimum level of QoS statistics by achieving a balance between conservation and consumption, one can define an appropriate reward function to achieve one's objective. Here, we define a reward function that incorporates both QoS statistics and some measure of the amount of energy available to the system to balance the QoS requirements and the energy needs of the system. Since the consumption rate u_l is a function of the duty cycle q, we can use the QoS statistics derived in Section III to derive a reward for each state in the MDP.
   In this work, we examine two different reward functions. The first reward function is an n-dimensional reward column vector with the following constituent entries for the corresponding states l

$$W(l) = -w_\pi \pi(l) - w_\ell \ell(l)/\ell^*(l) - w_\rho \rho(l) + w_A A, \tag{13}$$

   for some arbitrary weights w_π, w_ℓ, w_ρ and $w_A \in \mathbb{R}^+$, and where $\ell^*(l) = B/(\lambda(u_l) + \lambda_0(u_l))$. This definition of the reward offers a high reward for a low loss probability, low latency, low energy consumption and high energy availability A. Note that since A is dependent on some transition matrix that governs the time evolution of the system, $W(l)$ is not strictly a function of only the system state l, but is a non-stationary function that varies with time.
   The second reward function is a similar column vector with constituent entries in (14) involving

$$W(l) = \begin{cases} w_+ & \pi(l) < \pi_0, \ell(l) < \ell_0, \rho(l) < \rho_0, \\ & \quad A \in A_0 = [A_{0,-}, A_{0,+}) \\ w_- & \text{otherwise} \end{cases} \tag{14}$$

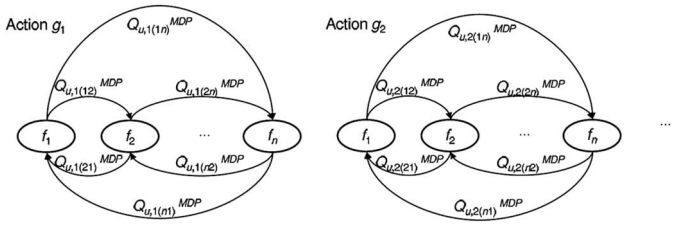


Fig. 4. MDP state and action spaces. $\mathcal{Q}_{u,s(ij)}^{MDP}$ refers to the (i, j)-th entry of the transition rate matrix corresponding to the s-th action.

   the latency and loss thresholds π_0 and ℓ_0 defined earlier in Section III, as well as analogous power consumption and availability thresholds ρ_0 and A_0. In this case, the reward takes one of two discrete values w_+ or w_- instead of a continuous spectrum of values in the earlier example.
   The continuous function (13) enables users to finetune the balance between energy availability and QoS requirements, while the thresholding function (14) enables users who are aware of the thresholds that the system is required to satisfy, such as network engineers, to provide a clear system input.
4) Quality: Last but not least, we associate each state f_l and action g_s with a quality $Q_{m,ls}$.

In Fig. 4, we use a state machine diagram to illustrate the state and action spaces of the MDP.

B. Implementing the MDP

We begin our MDP by selecting a suitable initial guess for the quality matrix $Q_m = [Q_{m,ls}]$, some initial state f_0, and a suitable initial CTMC transition matrix $\mathcal{Q}_{u,a}^{MDP} = \mathcal{Q}_{u,0}^{MDP}$. Then, we evolve our MDP as follows: based on some tuning parameter $0 < \epsilon_t < 1$, we select with probability ϵ_t the action $g_s = g_f \in \mathcal{G}$ with the highest value in the quality matrix Q_m for the corresponding state f_0, or with probability $1 - \epsilon_t$ a random action $g_s = g_r \in \mathcal{G}$. Based on this action, we select the f_0-th row in the transition matrix Q_{u,g_s} corresponding to the action g_s, and replace the f_0-th row in our actual CTMC transition matrix $\mathcal{Q}_{u,a}^{MDP}$ with this row. Then, we evolve the system based on $\mathcal{Q}_{u,a}^{MDP}$. When the system evolves to a new state f_1 based on this transition matrix, the reward associated with the new state $W(f_1)$ is computed using the value of A corresponding to the current matrix $\mathcal{Q}_{u,a}^{MDP}$. Next, the appropriate entry in the quality matrix is computed using the Q-learning method [44]

$$Q_{m,0s,\text{new}} = (1-\mu) Q_{m,0s,\text{old}} + \mu \left(W(f_1) + \gamma \max_{g_{s'}} Q_{m,1s',\text{old}} \right) \tag{15}$$

for some learning rate $0 < \mu < 1$ and some discount factor $\gamma \in [0, 1)$. The next action is then selected using the entries of the updated quality matrix, and the entire process is repeated for the entire time evolution process. The intention is to maximize the following expectation value over all sample paths

$$W_r = \mathbb{E}\left[\int_0^\infty \gamma^t W(f(t)) dt \right] \tag{16}$$

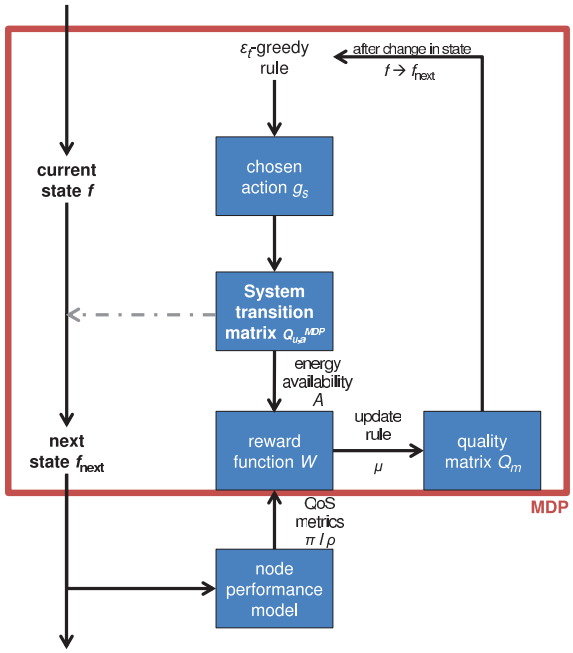


Fig. 5. MDP dynamics. At all times, the system evolves based on knowledge of the current state f and the system transition matrix $Q_{u,a}$. A new action is selected whenever the system undergoes a state transition.

where the state of the process at time t is $f(t)$. An optimal solution will attain the maximum possible expected discounted reward W_r. In our adaptive framework, we aim to maximize W_r for the system under consideration by tuning ϵ_t and μ, and choosing an appropriate set of actions $\mathcal{G}$. Under this framework, we believe the CTMC transition matrix $\mathcal{Q}_{u,a}^{MDP}$ increases its optimality over time in the long run and is able to respond to changes in the system parameters.

In Fig. 5, we use a flow diagram to illustrate the reward generation, quality update and decision-making process in the MDP. We also summarize the process in Algorithm 1.

Note that under the assumption of Markovian transitions between harvesting rates, we can choose to either establish a different state space, action space, quality matrix and instantaneous transition matrix for each harvesting rate, or we can choose to evolve a single MDP and simply let it respond to the harvesting rate variation with a time-varying reward (in particular the variation in A). The first option is more rigorous and is likely to provide better convergence, but the second option could be more convenient and less intensive to implement. Under the assumption of a smooth variation of the harvesting rate, we can evolve the MDP as it is and allow it to respond to harvesting rate variations with a time-varying reward, provided the MDP convergence timescale is shorter than the harvesting rate variation timescale.

VII. Simulation Results

In this Section, we present some results obtained using Monte Carlo simulations of a single node involving the simultaneous time evolution of the CTMCs describing the network and battery states of the node, and of the MDP describing

Algorithm 1. Procedure for solving MDP

Input: Initial guess for Q_m, initial state $f = f_0$, initial CTMC transition matrix $\mathcal{Q}_{u,a}^{MDP} = \mathcal{Q}_{u,0}^{MDP}$

Output: Instantaneous (actual) CTMC transition matrix $\mathcal{Q}_{u,a}^{MDP}$, Instantaneous state f

initialization;

while *node is functioning* **do**

- Select action g_s;
- With probability ϵ_t, determine if optimal action selection is triggered;
- **if** *optimal action selection triggered* **then**
  - Select g_s with highest corresponding value in Q_m for corresponding state f;
- **else**
  - Select random g_s;
- Update $\mathcal{Q}_{u,a}^{MDP}$ based on g_s;
- Evolve system in time based on $\mathcal{Q}_{u,a}^{MDP}$ until state f changes to f_{next};
- Compute reward $W(f_{\text{next}})$ and update corresponding entry in Q_m;
- Repeat loop

the evolution of the duty cycle of the node. We first describe the parameters and the MDP formulation used in the simulation. We then discuss the impact of the reinforcement learning parameters, and evaluate the performance of our algorithm for both the continuous and discrete reward functions highlighted in Section VI.

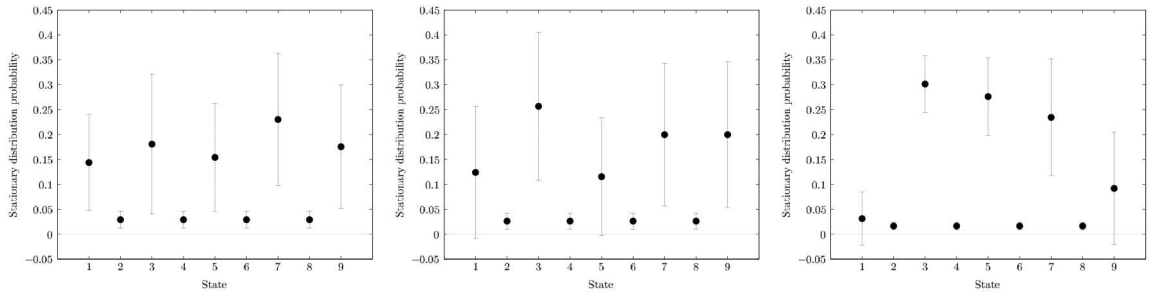
A. Simulation set-up

The network parameters used in our simulations are described in Table I. For simplicity, we consider the case where the node in question does not generate its own packets. Also, we first consider the case where the link quality between the nodes in the network is always good (so there is no difference between the Retransmissions and No Retransmissions schemes), and where the energy harvesting rate adopts a constant value of 0.3 energy units per unit time for the duration of the learning. (We relax the link quality and constant harvesting rate assumptions in Sections VII-E and VII-F respectively.) For the chosen set of network parameters, this bounds the possible battery evolution rates between about -0.03 and 0.07 energy units per unit time. Finally, we assign our battery a capacity of 0.05 energy units.

For our MDP, we consider a relatively small state space with 9 states, involving the regularly-spaced duty cycles $\{0.1, 0.2, \ldots, 0.9\}$, as well as a relatively small action space with 5 actions, to make the conclusions we derive from our simulations clearer. The infinitesimal generators corresponding to the five actions are detailed in Appendix B, and have corresponding stationary distributions that are centred around states 1, 3, 5, 7 and 9 respectively. In all trials, we initialize our MDP with a randomly chosen state, an all-zero quality matrix, and a state transition matrix where all inter-state transitions have a rate of 0.1 transitions per unit time, and force the MDP to select an action immediately.

TABLE I
NETWORK PARAMETERS

Notation	Description (units)	Value
T	Average cycle time (time units)	2
$\lambda_g = \lambda_b$	Average received-packet rate from nearby nodes (per time unit)	0.5
B	FIFO buffer size (packets)	10
$\theta_g = \theta_b$	Intensity of probing (per time unit)	3
$\mathcal{P}_{asleep}$	Power consumption of a node in the asleep state (energy units per time unit)	0.01
$\mathcal{P}_{awake}$	Power consumption of a node in the awake state (energy units per time unit)	0.1
$\mathcal{P}_{probe}$	Power consumption of probing mechanism (energy units per time unit)	0.2
ε_{tx}	Energy consumed to transmit a single data packet (energy units)	0.4



(a) $\epsilon_t = 0.1$ (b) $\epsilon_t = 0.5$ (c) $\epsilon_t = 0.9$

Fig. 6. Stationary distribution based on the transition rate matrix $\mathcal{Q}^{\text{MDP}}_{u,a}$ for the 9 MDP states averaged over 50 trials each for different values of ϵ_t. Here, $\mu = 0.1$ and $\gamma = 0.995$. In addition, we use the reward function (13) and set all weights to 1.

B. Reinforcement learning parameters

We highlight the effects of varying the learning rate μ, the exploitation probability ϵ_t, and the discount factor γ. In particular, we note the trade-off between decreasing the response time of the system and maintaining the stability of the convergence of the learning curves, and make the observation that the reinforcement learning algorithm used in our simulations outperforms a random decision-making scheme.

1) Effects of varying learning rate μ: We observed that a higher μ increases the learning ability of the system by decreasing the convergence time, but results in higher-frequency and larger fluctuations in the learned quality values.

2) Effects of varying exploitation probability ϵ_t: We observed that a higher ϵ_t decreases the convergence time of the system at the expense of higher-frequency and larger fluctuations in the learned quality values, the effect of which is seen in Fig. 6. Here, the distribution deviates more strongly from a uniform distribution for the odd-numbered states, which is what we would expect in a system that selects its actions randomly. In addition, as ϵ_t increases, the standard deviations of the probabilities decrease, suggesting convergence towards a desirable stationary distribution. By choosing a non-zero ϵ_t, we obtain a scheme that outperforms a random decision-making policy.

3) Effects of varying discount factor γ: Without a sufficiently large γ, the randomness inherent in a MDP, and especially in the reward function we selected due to the constantly varying nature of A, may impede the learning of the system. We observed that only with a sufficiently large γ does effective learning take place in the system.

C. Performance of algorithm: Continuous reward function

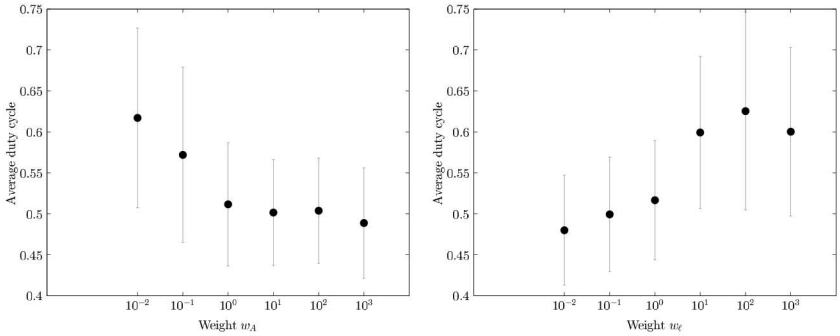
The performance of our algorithm depends on our system objectives, as well as the reward function implemented in our MDP. An appropriate choice of the reward function can shift the behaviour of the system. Here, we consider the reward function (13), and look at the effects of varying the weights.

1) Effects of energy availability on system: We expect that in a system that places great emphasis on energy availability, the average duty cycle will be low in order to ensure the battery does not expend all its energy; conversely, in a system that places little emphasis on energy availability, the average duty cycle will then be governed by the QoS metrics, which are likely to push the duty cycle higher to ensure effective transmission. This is validated by the results in Fig. 7a.

2) Effects of latency on system: We similarly expect that when emphasis on latency is high, the average duty cycle will be high to maintain the QoS standards; conversely, when emphasis on latency is low, the average duty cycle will then be lower to conserve energy and meet the power consumption and energy availability requirements. This intuition is validated by the results in Fig. 7b.

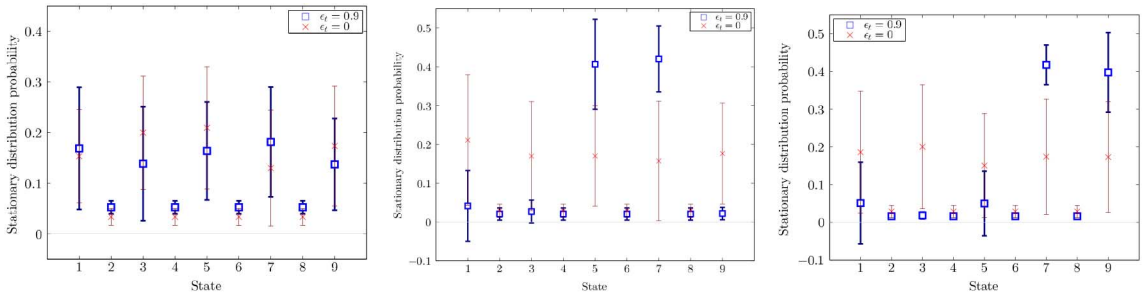
D. Performance of algorithm: Reward function with thresholding

Figure 8 shows the effect of various QoS thresholds on the selection of actions, as well as the comparison of the random and ϵ_t-greedy approaches, the latter of which we highlighted in our algorithm above. The three cases of thresholds we consider are: Case I: $\{\rho_0 = 0.3, \ell_0 = 5, \pi_0 = 0.01\}$; Case



(a) Varying w_A (b) Varying w_ℓ

Fig. 7. Plot of average duty cycle versus (a) w_A, holding all other weights constant at 1 and (b) w_ℓ, holding all other weights constant at 1, using (13) and averaging over 50 trials for each w_A and w_ℓ respectively. Here, $\epsilon_t = 0.5$, $\mu = 0.1$ and $\gamma = 0.995$.



(a) Case I (b) Case II (c) Case III

Fig. 8. Comparison of the ϵ_t-greedy approach highlighted in our algorithm above ($\epsilon_t = 0.9$, blue) with a random approach ($\epsilon_t = 0$, red) using the average stationary probability of each state over 20 trials for three different cases of thresholds. The more probable states obtained from our algorithm (in blue) agree well with the optimal duty cycles predicted in Fig. 9.

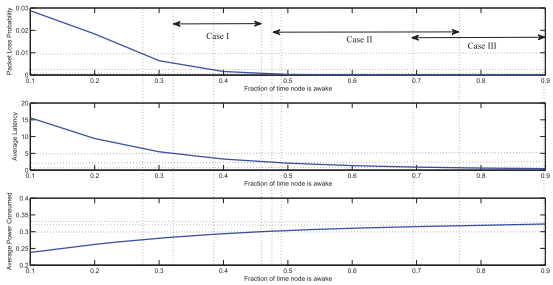


Fig. 9. Optimal sets of duty cycles for each case without considering the energy availability A.

II: $\{\rho_0 = 0.32, \ell_0 = 5, \pi_0 = 0.01\}$; Case III: $\{\rho_0 = 0.33, \ell_0 = 0.91, \pi_0 = 4\text{E-}5\}$. Here, we do not consider the effects of energy availability by taking $A_0 = [0, 1)$. Under these circumstances, we observe that the RL thresholding algorithm converges. The scenario when the energy availability is taken into account is considered in the next subsection.

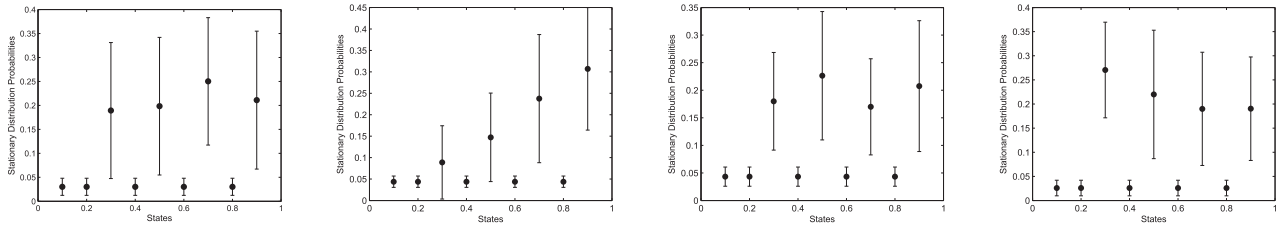
Figure 9 shows the optimal sets of duty cycles for the three cases in order to satisfy the QoS thresholds. In Case I, the corresponding optimal state is 4. In Case II, the corresponding optimal states are 5, 6 and 7. In Case III, the corresponding optimal states are 7, 8 and 9. This agrees with the results shown in Fig. 8 when $\epsilon_t = 0.9$, keeping in mind that our action space tends to direct the system towards the odd-numbered states only. When $\epsilon_t = 0$, the scheme reduces to a random approach where the states are chosen randomly and the stationary probabilities of states 1, 3, 5, 7 and 9 are uniform.

1) Effects of energy availability: As seen in the previous subsection, the QoS thresholds can help to choose a set of optimal duty cycles. However, the energy availability depends on the attainment of equilibrium by the whole system, and the derivation of the optimal set of duty cycles is not as straightforward.

a) Simulation settings: In this section, we modify the learning rate μ to be inversely proportional to the total time spent in the current state [32] in order to handle the variation in the energy availability. In addition, we select a subset of the action space in which the system's steady state probabilities are centred around states 3, 5, 7 and 9 to better illustrate the effects of varying A_0. The convergence of the algorithm with the presence of the energy availability term is not as fast with the learning rates used in the previous sections. Hence, we use a learning rate that decreases with total time elapsed in the current state (including past transitions) to speed up convergence.

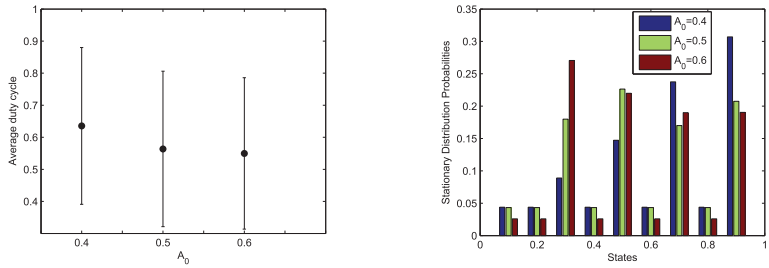
b) Analysis of steady state probabilities: Fig. 10a demonstrates that when energy availability is not considered and when all the duty cycles satisfy the given QoS thresholds, the steady state probabilities learned are almost uniform with high standard deviation. This changes in Figs. 10b to 10d, where we see that a low A_0 selects higher duty cycles and a high A_0 selects lower duty cycles on average.

c) Summary: For greater clarity, the results above are summarised in Fig. 11. These graphs demonstrate that an increase in A_0 decreases the average duty cycle and promotes the selection of actions favouring states that correspond to lower duty cycles.



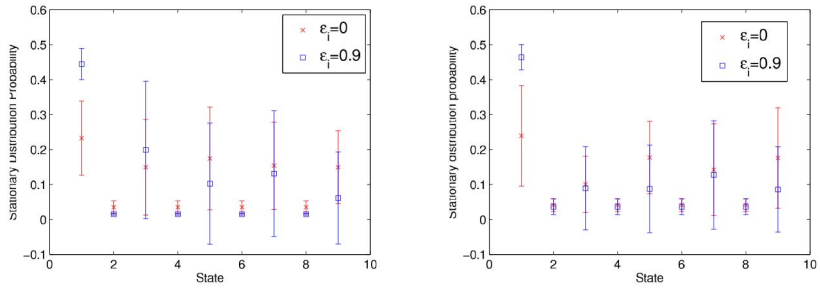
(a) $A_0 \in [0, 1)$ (b) $A_0 \in [0.4, 0.45)$ (c) $A_0 \in [0.5, 0.55)$ (d) $A_0 \in [0.6, 0.65)$

Fig. 10. Graphs of steady state probabilities averaged over 20 trials, for different ranges of A_0. Here, $\pi_0 = 0.02$, $\ell_0 = 9$, $\rho_0 = 0.33$, $\epsilon_t = 0.9$ and $\gamma = 0.995$.



(a) Average duty cycle for different A_0 (b) Average steady state probabilities for different A_0

Fig. 11. Graphs of (a) the duty cycle for different ranges of A_0 plotted against the median of A_0 and (b) the steady state probabilities for different ranges of A_0, both averaged over 20 trials. Here, $\pi_0 = 0.02$, $\ell_0 = 9$, $\rho_0 = 0.33$, $\epsilon_t = 0.9$ and $\gamma = 0.995$.



(a) Case I (Retransmissions) (b) Case II (No Retransmissions)

Fig. 12. Comparison of the ϵ_t-greedy approach in our algorithm ($\epsilon_t = 0.9$) with a random approach ($\epsilon_t = 0$) using the average stationary probability of each state over 10 trials for two different cases of thresholds. (a) Retransmissions scheme: $\lambda_g = \lambda_b = 0.5$. (b) No Retransmissions scheme: $\lambda_g = 0.5$, $\lambda_b = 0.3$.

2) Remarks: As we mentioned earlier, the continuous reward function (13) enables finetuning of the balance between energy availability and QoS requirements, while the thresholding reward function (14) enables clear demarcations of the boundaries of the system, provided the intersection of the threshold requirements is physically achievable. Our results demonstrate that the convergence characteristics of both reward mechanisms are comparable and can simultaneously take into consideration short-term QoS requirements and long-term energy availability standards.

E. Varying link quality

We now consider the effects of variable link quality as discussed in the system model in Section II. Here, we use the thresholding reward function given by (14).

Figure 12 compares the random and ϵ_t-greedy approaches, for the Retransmissions (Case I) and No Retransmissions (Case II) schemes. The corresponding optimal state for both Cases is 1, and we demonstrate in Fig. 12 that our algorithm is able to learn this optimal condition. Also, the RL algorithm converges for both transmission schemes.

F. Varying harvesting rate

We now consider the effects of variable harvesting rate to simulate environmental fluctuations. Here, we again use the thresholding reward function (14) and the same simulation settings as Section VII-D1a.

1) Simulation setup: In the simulations to be described, we used 4 different values of the harvesting rate $h_{\text{arr}} =$ [0.2, 0.2875, 0.3, 0.4] where the lowest rate corresponds to zero energy availability ($A = 0$) and the largest rate corresponds to maximum energy availability ($A = 1$). In addition, we define the harvesting rate to vary in a cyclical manner such that h_{arr} is uniformly distributed over time.

2) Simulation Results: We used a time-varying learning rate similar to Section VII-D1a to handle the changing environment. This ensures that the RL algorithm converges.

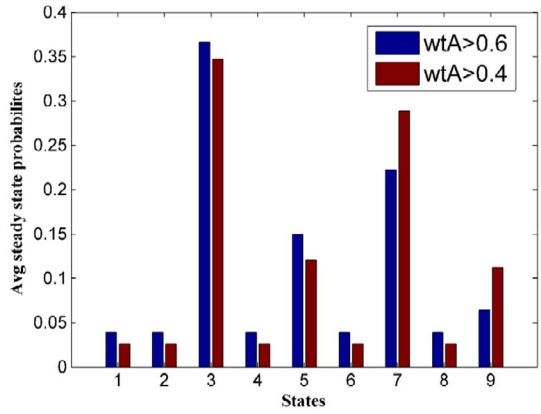


Fig. 13. Steady state probabilities for two different sets of energy availability thresholds A_0 over an average of 5 trials. In Case I (red), $A_0 = [0.4, 1]$, and in Case II (blue), $A_0 = [0.6, 1]$.

From Fig. 13, it is evident that when $A_0 = [0.6, 1]$, the lower duty cycle states are preferred, in contrast to the case when $A_0 = [0.4, 1]$, where higher duty cycle states are preferred. Thus, our algorithm is cognizant of the requirements in our system even under varying environmental conditions.

VIII. Conclusion

In this paper, we develop an adaptive duty cycling scheme in wireless sensor networks that takes into account both the energy supply dynamics and application-level QoS requirements at the same time. Continuous Time Markov Chain (CTMC) models are used to derive analytical expressions for these QoS metrics - such as latency, loss probability and average energy consumption - as well as for the energy availability of the system, which offers a probabilistic measure of the ability of the battery to maintain a non-zero energy level on average. We then establish a reward framework that allows the network to tune the relative importance of the energy availability and the QoS requirements, and we perform numerical simulations to verify our model. We implement a reinforcement learning algorithm that converges to a desirable solution quickly and with lower computational complexity than a completely optimal algorithm, and show that our adaptive scheme performs better than a random scheme. With the quick convergence of our algorithm, we enable the possibility of the system adapting to changes in the energy supply dynamics or the network transmission statistics that take place at timescales larger than the convergence time of the learning curve. We intend to extend this work by looking at update rules and threshold functions that enhance the decision-making ability of the system, and hopefully further increase the responsiveness of the system to unexpected externalities.

Appendix A
Proof of Lemma 3

A. Transient behaviour of battery

Let $F(t, x, j; y, i) = P_r(X(t) \leq x, Z(t) = j | X(0) = y, Z(0) = i)$. $F(t, x, j; y, i)$ gives the cumulative transition probability that the battery level $X(t)$ is at most x at time t and that the battery is in state j, given that the battery was originally in state i with battery level y. It can be shown that the cumulative transition probability mn-by-mn matrix $F(t, x; y) = [f_{ij}]$ satisfies the equations

$$\frac{\partial F(t, x; y)}{\partial t} + \frac{\partial F(t, x; y)}{\partial x} D = F(t, x; y) \mathbb{Q}_e^{MDP} \tag{17}$$

for each $x \in [0, C]$ and $y \in [0, C]$, with boundary conditions

$$\begin{aligned} F(t, 0, j; y, i) &= 0, \quad \text{if } r_j > 0, \\ F(t, C, j; y, i) &= \pi_{ij}(t), \quad \text{if } r_j < 0. \end{aligned} \tag{18}$$

B. Steady-state behaviour of battery

As t goes to infinity, the limits of (17) and (18) become, for the mn-dimensional row vector $F(x)$ with entries $F(x, j)$,

$$\frac{dF(x)}{dx} D = F(x) \mathbb{Q}_e^{MDP} \tag{19}$$

for each $x \in [0, C]$, with boundary conditions

$$\begin{aligned} F(0, j) &= 0 \qquad \text{if } r_j > 0, \\ F(C, j) &= \pi_j \qquad \text{if } r_j < 0. \end{aligned} \tag{20}$$

We see that

$$F' = F(\mathbb{Q}_e^{MDP} D^{-1}), \tag{21}$$

which leads us to guess the solutions $F(x) = e^{\lambda x} \phi$ where λ is a scalar and ϕ is an mn-dimensional row vector. It can be shown that the general solution to (21) is given by $F(x) = \sum_{i \in T} a_i e^{\lambda_i x} \phi_i$, where λ_i are the generalized eigenvalues and ϕ_i the generalized eigenvectors of the equation

$$\phi_i (\lambda_i D - \mathbb{Q}_e^{MDP}) = 0 \tag{22}$$

or

$$((\mathbb{Q}^{MDP})_e^T - \lambda_i D^T) \phi_i^T = 0. \tag{23}$$

In other words, λ_i and ϕ_i^T are the eigenvectors of $(D^{-1})^T (\mathbb{Q}^{MDP})_e^T = ((\mathbb{Q}^{MDP})_e D^{-1})^T$.

The coefficients a_i are given by the solutions to

$$\begin{aligned} \sum_{i \in T} a_i \phi_i(j) &= 0 \qquad \text{if } j \in T^+, \\ \sum_{i \in T} a_i \phi_i(j) e^{\lambda_i C} &= \pi_j \qquad \text{if } j \in T^-, \end{aligned} \tag{24}$$

where T^+ and T^- contain the elements of T where the corresponding rate r_j is positive and negative respectively. Note that keeping all else constant, a higher C results in lower a_i and thus a larger A.

APPENDIX B
ACTION SPACE OF MDP USED FOR SIMULATIONS

The infinitesimal generators corresponding to the five actions are such that they encourage the system to transition to states 1, 3, 5, 7 and 9 respectively. An action that transitions the system to some state i has an infinitesimal generator where state i has a transition rate of 0.1 to all other states, and all other states have transition rates of 0.9 to state i and 0.1 to all other states. For example, for action 1, the infinitesimal generator is (25), shown at the bottom of the page.

REFERENCES

[1] A. Kottas, Z. Wang, and A. Rodrguez, "Spatial modeling for risk assessment of extreme values from environmental time series: A Bayesian nonparametric approach," *Environmetrics*, vol. 23, no. 8, pp. 649–662, 2012.

[2] C. Fonseca and H. Ferreira, "Stability and contagion measures for spatial extreme value analyses," *arXiv preprint arXiv:1206.1228*, 2012.

[3] J. P. French and S. R. Sain, "Spatio-temporal exceedance locations and confidence regions," *Ann. Appl. Statist.*, vol. 7, no. 3, pp. 1421–1449, 2013.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[5] W. K. Seah, Z. A. Eu, and H.-P. Tan, "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-survey and challenges," in *Proc. 1st Int. Conf. Wireless Commun. Veh. Technol. Inf. Theory Aerosp. Electron. Syst. Technol.*, 2009, pp. 1–5.

[6] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2005, pp. 457–462.

[7] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman, "Challenge: Ultra-low-power energy-harvesting active networked tags (EnHANTs)," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 253–260.

[8] C. Bergonzini, D. Brunelli, and L. Benini, "Algorithms for harvested energy prediction in batteryless wireless sensor networks," in *Proc. 3rd Int. Workshop Adv. Sensors Interfaces*, Jun. 2009, pp. 144–149.

[9] D. Hasenfratz, A. Meier, C. Moser, J.-J. Chen, and L. Thiele, "Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks," in *Proc. IEEE Int. Conf. Sensor Netw. Ubiquitous Trustworthy Comput. (SUTC)*, Jun. 2010, pp. 19–26.

[10] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surv. Tuts.*, vol. 13, no. 3, pp. 443–461, Third Quart. 2011.

[11] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 6, pp. 1–38, Sep. 2007.

[12] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, and V. Raghunathan, "Adaptive duty cycling for energy harvesting systems," in *Proc. Int. Symp. Low Power Electron. Des.*, 2006, pp. 180–185.

[13] C. Vigorito, D. Ganesan, and A. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," in *Proc. IEEE Sensor Mesh Ad Hoc Commun. Netw. (SECON)*, 2007, pp. 21–30.

[14] Y. Gu, T. Zhu, and T. He, "ESC: Energy synchronized communication in sustainable sensor networks," in *Proc. IEEE Int. Conf. Netw. Protocols (ICNP)*, 2009, pp. 52–62.

[15] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang, "Leakage-aware energy synchronization for wireless sensor networks," in *Proc. 7th Int. Conf. Mobile Syst. Appl. Serv. (MobiSys)*, 2009, pp. 319–332.

[16] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, "Optimal energy management policies for energy harvesting sensor nodes," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1326–1336, Apr. 2010.

[17] C. Moser, J.-J. Chen, and L. Thiele, "An energy management framework for energy harvesting embedded systems," *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, vol. 6, no. 2, p. 7, 2010.

[18] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design Test Comput.*, vol. 18, no. 2, pp. 62–74, Mar./Apr. 2001.

[19] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1853–1865, Sep. 2013.

[20] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 16–27, Oct. 2000.

[21] W. B. Heinzelman, A. L. Murphy, H. S. Carvalho, and M. A. Perillo, "Middleware to support sensor network applications," *IEEE Netw.*, vol. 18, no. 1, pp. 6–14, Jan./Feb. 2004.

[22] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[23] J. Lei, R. Yates, and L. Greenstein, "A generic model for optimizing single-hop transmission policy of replenishable sensors," *IEEE Trans. Wireless Commun.*, vol. 8, no. 2, pp. 547–551, Feb. 2009.

[24] A. Seyedi and B. Sikdar, "Energy efficient transmission strategies for body sensor networks with energy harvesting," *IEEE Trans. Commun.*, vol. 58, no. 7, pp. 2116–2126, Jul. 2010.

[25] H. Li, N. Jaggi, and B. Sikdar, "Relay scheduling for cooperative communications in sensor networks with energy harvesting," *IEEE Trans. Wireless Commun.*, vol. 10, no. 9, pp. 2918–2928, Sep. 2011.

[26] A. Sultan, "Sensing and transmit energy optimization for an energy harvesting cognitive radio," *IEEE Wireless Commun. Lett.*, vol. 1, no. 5, pp. 500–503, Oct. 2012.

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

[28] R. Chan *et al.*, "Adaptive duty cycling in sensor networks via continuous time Markov chain modeling," in *Proc. IEEE Int. Conf. Commun. (ICC'15)*, Jun. 2015, pp. 6669–6674.

[29] V. G. Kulkarni, "Fluid models for single buffer systems," in *Frontiers in Queueing: Models and Applications in Science and Engineering*. Boca Raton, FL, USA: CRC Press, 1997, pp. 321–338.

[30] G. L. Jones, P. G. Harrison, U. Harder, and T. Field, "Fluid queue models of battery life," in *Proc. IEEE 19th Int. Symp. Model. Anal. Simul. Comput. Telecommun. Syst. (MASCOTS)*. 2011, pp. 278–285.

[31] N. Gautam and A. Mohapatra, "Efficiently operating wireless nodes powered by renewable energy sources," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 8, pp. 1706–1716, Aug. 2015.

[32] A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

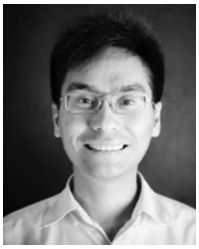
[33] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[34] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, 1960.

[35] E. O. Elliot, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 1977–1997, 1963.

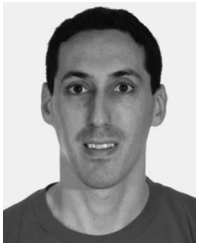
$$\mathcal{Q}_{u,1}^{MDP} = \begin{bmatrix} -0.8 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.9 & -1 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 \\ 0.9 & 0.0143 & -1 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 \\ 0.9 & 0.0143 & 0.0143 & -1 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 \\ 0.9 & 0.0143 & 0.0143 & 0.0143 & -1 & 0.0143 & 0.0143 & 0.0143 & 0.0143 \\ 0.9 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & -1 & 0.0143 & 0.0143 & 0.0143 \\ 0.9 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & -1 & 0.0143 & 0.0143 \\ 0.9 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & -1 & 0.0143 \\ 0.9 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & 0.0143 & -1 \end{bmatrix}. \quad (25)$$

[36] P. Poggi, G. Notton, M. Muselli, and A. Louche, "Stochastic study of hourly total solar radiation in corsica using a Markov model," *Int. J. Climatol.*, vol. 20, no. 14, pp. 1843–1860, 2000.
[37] O. Kwon, Y.-J. Yoon, S. K. Moon, H.-J. Choi, and J. H. Shim, "Estimation of Singapores hourly solar radiation using Hybrid-Markov transition matrices method," *Int. J. Precis. Eng. Manuf.*, vol. 14, no. 2, pp. 323–327, 2013.
[38] R. Jayaraman and D. L. Maskell, "Temporal and spatial variations of the solar radiation observed in Singapore," *Energy Procedia*, vol. 25, pp. 108–117, 2012.
[39] Z. Liu and I. Elhanany, "Rl-MAC: A QOS-aware reinforcement learning based MAC protocol for wireless sensor networks," in *Proc. IEEE Int. Conf. Netw. Sens. Control.*, 2006, pp. 768–773.
[40] Z. Liu and I. Elhanany, "Rl-MAC: A reinforcement learning based MAC protocol for wireless sensor networks," *Int. J. Sensor Netw.*, vol. 1, no. 3, pp. 117–124, 2006.
[41] G. Ghidini and S. K. Das, "An energy-efficient Markov chain-based randomized duty cycling scheme for wireless sensor networks," in *Proc. 31st Int. Conf. Distrib. Comput. Syst. (ICDCS'11)*, 2011, pp. 67–76.
[42] M. Bkassiny, S. K. Jayaweera, and K. A. Avery, "Distributed reinforcement learning based mac protocols for autonomous cognitive secondary users," in *Proc. IEEE 20th Annu. Wireless Opt. Commun. Conf. (WOCC'11)*, 2011, pp. 1–6.
[43] G. Ghidini and S. K. Das, "Energy-efficient Markov chain-based duty cycling schemes for greener wireless sensor networks," *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, vol. 8, no. 4, p. 29, 2012.
[44] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, 1992.



Wai Hong Ronald Chan received the B.Sc. degree in engineering from Massachusetts Institute of Technology, Cambridge, MA, USA, in 2014. He is currently a Graduate Student at the Department of Mechanical Engineering, Stanford University, Stanford, CA, USA. From 2014 to 2015, he was a Research Engineer with the Institute of Infocomm Research (I2 R) under the Sense and Sense-Abilities Programme. His research interests include transport phenomena, multiphysics modelling, network optimization, and probabilistic network interactions.

Pengfei Zhang received the Bachelor of Engineering in electrical and electronic engineering and Ph.D. degree from Nanyang Technological University, Singapore, in 2010 and 2015, respectively. He is currently working as a Research Scientist with the Institute of Infocomm Research (I2 R) under the Sense and Sense-Abilities Programme. His research interests include energy efficient clustering algorithms, energy harvesting wireless sensor networks (EH-WSNs), security intelligence, and statistical modelling in WSNs.



Ido Nevat received the B.Sc. degree in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, and the Ph.D. degree in electrical engineering from the University of NSW, Sydney, N.S.W., Australia, in 1998 and 2010, respectively. From 2010 to 2013, he was a Research Fellow with the Wireless and Networking Technologies Laboratory, CSIRO, Australia. Currently, he is a Research Scientist with the Institute of Infocomm Research (I2R), Singapore. His research interests include statistical signal processing and Bayesian statistical modeling.

Sai Ganesh Nagarajan received the Undergraduate degree in computer engineering from the National University of Singapore, Singapore. He is currently working as a Research Engineer with the Institute of Infocomm Research (I2R). His research interests include statistical modeling, neural networks, reinforcement learning, and dynamical systems.

Alvin C. Valera received the Bachelor of Science degree (*cum laude*) in computer engineering from the University of the Philippines (Diliman), Quezon City, Philippines, the Master of Science in computer science and Doctor of Philosophy in electrical and computer engineering from the National University of Singapore, Singapore, in 1998, 2003, and 2015, respectively. He is currently a Research Fellow with the School of Information Systems, Singapore Management University, Singapore. He has worked on research and development projects related to wireless networking including mobile ad hoc networks, underwater networks, and sensor networks. His research interests include IoT network management and, optimization and control of wireless sensor networks powered by ambient energy harvesting.

Hwee-Xian Tan received the Ph.D. degree from the School of Computing, National University of Singapore, Singapore, in 2011. She is a Research Scientist with the SMU-TCS iCity Laboratory, Singapore Management University (SMU), Singapore. She is part of the Smart Homes and Intelligent Neighbors to Enable Seniors (SHINESeniors) Project Team. Her current research interests include smart city applications, Internet of Things (IoT), wireless networking and technology, and embedded systems.

Natarajan Gautam received the B.Tech. degree from Indian Institute of Technology, Madras, India, the M.S. and Ph.D. degrees in operations research from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. He is a Professor with the Department of Industrial and Systems Engineering, Texas A& M University, College Station, TX, USA. Prior to joining Texas A& M University in 2005, he was a Industrial Engineering Faculty with Pennsylvania State University, State College, PA, USA, for eight years. His research interests include modeling, analysis and performance evaluation of stochastic systems with special emphasis on optimization and control in computer, and telecommunication and information systems. He is an Associate Editor for the *INFORMS Journal on Computing*, *IIE Transactions*, and *OMEGA*.